10-23-2014 12:00 AM

Censored Time Series Analysis

Nagham Muslim Mohammad
The University of Western Ontario

Supervisor
Dr.A. Ian McLeod
The University of Western Ontario

Graduate Program in Statistics and Actuarial Sciences
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

CENSORED TIME SERIES ANALYSIS
(Thesis format: Monograph)

by

Nagham Muslim Mohammad

Graduate Program in Statistics and Actuarial Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

Environmental data is frequently left or right censored. This is due to the fact that the correct value for observed values that are below or above some threshold or detection point are inaccurate so that it is only known for sure that the true value is below or above that threshold. This is frequently important with water quality and air quality time series data. Interval censoring occurs when the correct values of the data are known only for those values falling above some lower threshold and below some upper threshold. Censoring threshold values may change over time, so multiple censor points are also important in practice. Further discussion and examples of censoring are discussed in the first chapter. A new dynamic normal probability plot for censored data is described in this chapter.

For some environmental time series the effect of autocorrelation is negligible and we can treat the data or often the logged data as a random sample from a normal population. This case has been well studied for more than half a century and the work on this is briefly reviewed in the second chapter. The second chapter also contains a new derivation and a new algorithm based on the EM algorithm for obtaining the maximum likelihood estimates of the mean and variance from censored normal samples. A new derivation is also given for the observed and expected Fisher information matrix.

In chapter three the case of autocorrelated time series is discussed. We show the close relationship between censoring and the missing value problem. A new quasi-EM algorithm for missing value estimation in time series is described and it efficacy demonstrated. This algorithm is extended to handle censoring in the general case of multiple censor points and interval censoring. When there is no autocorrelation, this algorithm reduces to the algorithm developed in Chapter 2. An application to water quality in the Niagara river is discussed.

Keywords: Time series analysis, censoring, environmetrics

Acknowledgements

First and foremost I would like to express my sincere gratitude to my supervisor Professor A. Ian McLeod for his invaluable guidance and generous support throughout the course of my study at Western. I would also like to thank my thesis examiners, Professors Dr. Hao Yu, Dr. Jiandong Ren, Dr. Jean-Marie Dufour and Dr. John Koval for carefully reading this thesis and helpful comments. I am also grateful to all faculty, staff and fellow students at the Department of Statistical and Actuarial Sciences for their encouragement. The author would also like to acknowledge the financial support of the Department of Statistical and Actuarial Sciences and the Faculty of Graduate Studies and the Board of the Ontario Graduate Scholarships in Science and Technology. Finally, I would like to thank my family for their patience and love that helped me to reach this point.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and literature review

In this chapter a review is given on the underlying methods that are developed in the later chapters. This thesis deals with what is known as Type 1 censoring. There is an extensive literature on this subject with most of the research focused on the random sample case. The monograph by Schneider [1986] focuses on censoring with normal random samples while the monograph of Cohen [1991] discusses the more general case of random samples from normal and non-normal distributions. Wolynetz [1979a,b, 1980, 1981] has implemented Fortran algorithms for censored normal samples and censored regression. The recent monograph of Helsel [2011] provides and extensive overview of recent research with a focus on computation and censored environmental data. The seminal paper of Kaplan and Meier [1958] developed a nonparametric method for fitting survival distributions with incomplete observations. Subsequently, the Kaplan-Meier curves have been widely used with times-to-event data. Very little has been done with censored time series and with the problem of fitting time series models to censored data. The paper by Park et al. [2007] is a notable exception. However as will be discussed later the algorithms and methods given in this paper are incorrect and not very useful. Hopke et al. [2001] discuss a data augmentation algorithm that is implemented to fit a non-stationary multivariate time series model to a time series of average weekly airborne particulate concentrations of twenty four variables. The model is essentially equivalent to a vectorized version of the ARIMA(0,1,1) model.

In Chapter 2, the EM algorithm is discussed for the simple case of estimation of the mean and variance in the censored time series model consists of a mean plus Gaussian white noise. This is equivalent to the well-known and much studied problem of censored samples from a normal distribution [Cohen, 1991, Schneider, 1986, Wolynetz, 1979a] We present a new derivation using the EM algorithm as well a new closed form expression for the information matrix for the mean and variance parameters. It is shown that in the censored case these parameters are not orthogonal. A new interactive normal probability plot for censored data is discussed. Several applications are given.

Chapter 3 develops a new Quasi-EM algorithm for fitting ARMA, stationary ARFIMA and other linear time series models to censored time series. It is shown that the missing value problem in time series model fitting may be regarded as a special and extreme case of censoring and it is demonstrated that our approximate Quasi-EM algorithm handles this case just as well as the standard exact treatment. The method is illustrated with an application.

Censoring with environmental data and time series

Pollution and the monitoring of toxic substances in rivers, lakes and in the atmosphere may gives rise to samples or to time series data that are censored due to the technology used to measure the quantity of the toxic substance. Typically a sample of water or air is taken and then laboratory analysis may be used to measure the amount or concentration of the toxic substance present. The monograph of Helsel [2011] discusses statistical methods for the analysis of censored environmental data and provides many interesting environmental datasets.

The type of censoring with environmental data is known as Type 1 censoring in contrast to Type 2 censoring in which the censored observations result from stopping an experiment at a preset time.

Consider the case $Z_i, i = 1, \ldots, n$ are independent and identically distribution and the distribution function $F(z)$ corresponds to the complete data. We do not observe $Z_1, \ldots, Z_n$ but instead we observe a censored version, denoted by $Y_i, i = 1, \ldots, n$. In the case of left-censoring $Y_i = \max(Z_i, c_i), i = 1, \ldots, n$ where c_i are the censor points or detection thresholds. Without loss of generality we can partition the sample $(Z_1, Z_2, \ldots, Z_n)$ into two parts $(Y_1, ..., Y_m)$ and $(Z_1, \ldots, Z_{n-m})$ where $Y_1, \ldots, Y_m$ correspond to the non-censored or complete observations and $Z_1, \ldots, Z_{n-m}$ correspond to the censored observations. The right-truncated distribution of the censored values is shown in Figure 1.1 below. Due to censoring the actual values of $Z_1, \ldots, Z_{n-m}$are not observed and in their place only $Y_i = c_i, i = m+1, \ldots, n$ is observed. Right censoring is defined by setting $Y_i = \min(Z_i, c_i), i = 1, \ldots, n$. An important special case is single-censoring when $c_1 = \ldots = c_n = c$. The single, left-censored normal distribution case is shown in Figure 1.1.

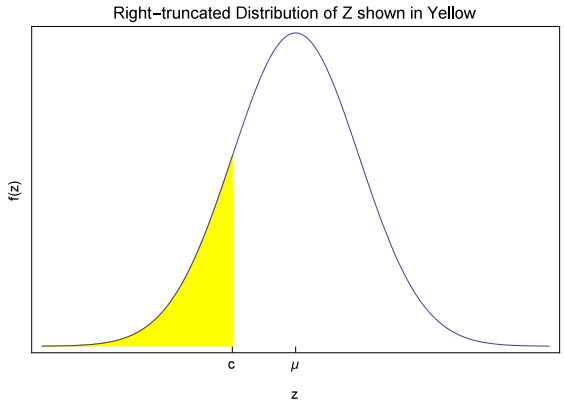

Figure 1.1: Left-censoring and the corresponding right-truncated distribution.

Left-censored data or time series often arise in water or air quality applications since the instruments may not be able to measure below or above some specified threshold, c. In this case the data may be left-censored, right-censored or if there are both upper and lower limits, interval censored. In lifetime data in engineering and medical sciences, right censoring of survival times is of interest [Lawless, 2003, §2.2] and we illustrate our new dynamic normal probability plot for censored data with an application to engine lifetimes that are right-censored.

In practice the detection level may change so that there are two or more detection levels. In the completely general left-censored case we assume that each observed value, $Y_i, i = 1, \ldots, n$

is paired with a threshold value $c_i^-, i = 1, \ldots, n$ so if the underlying latent process is $Z_i, i = 1, \ldots, n$ our observed process is $Y_i = \max\left(Z_i, c_i^-\right), i = 1, \ldots, n$ in the left-censored case. It is assumed that $c_i^-, i = 1, \ldots, n$ are known. In the singly-censored case, $c_i^- = c, i = 1, \ldots, n$. Similarly in the right censored case, given c_i^+, we assume the unobserved latent value Z_i and we observe $Y_i = \min\left(Z_i, c_i^+\right), i = 1, \ldots, n$. If the data is both left and right censored it is said to be interval censored. It is assumed that the censoring process that determines $c_i, i = 1, \ldots, n$ is independent of the data generation process that generates the underlying latent process, $Z_i, i = 1, \ldots, n$. See Helsel [2011, §3.2] for more discussion and examples of the pitfalls and bias created by insider or non-independent censoring. This is not an issue with most data collected by reputable government agencies such as Environment Canada.

Simulated example

A random sample from an N(0, 1) distribution of size $n = 60$ was generated and all values less than -0.5 were set to -0.5. A normal probability is a useful tool for preliminary data analysis and is shown for this data in Figure 1.2. We see that there were 13 censored values, so the censor rate is $13/60 \approx 22\%$. Later we will introduce a new dynamic normal probability plot that is useful for robust estimation and diagnostic checking.

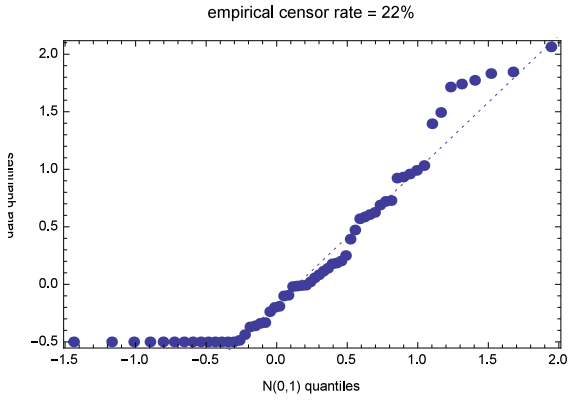

Figure 1.2: Simple normal probability plot of some left censored simulated N(0,1) data with censor point $c = -0.5, i = 1, \ldots, 60$.

Air quality example

Park et al. [2007] fit an AR(2) model, $\phi(B)(z_t - \mu) = a_t$, where $\phi(B) = 1 - \phi_1 B - \phi_2 B^2$, B is the backshift operator on t, μ is the series mean and a_t are the innovations which are assumed to be independent and identically distributed with a normal distribution with mean zero and variance σ_a^2, to a time series of hourly cloud ceiling heights in units of 100 meters. The time series had three missing values as well and were right censored and $c = c^+ = 12000$feet. For fitting a log transformation was used, so $c = 4.79$ log feet. The series was of length $n = 716$. The fitted model was an AR(2) with $\hat{\phi}_1 = 0.689 \pm 0.038$, $\hat{\phi}_2 = 0.173 \pm 0.038$, $\hat{\mu} = 4.129 \pm 0.236$ and $\hat{\sigma}_a = 0.877$. The observed censoring rate was 41.62%. This model was simulated and the time

series plot is shown in Figure 1.3. The observed censoring rate in the fitted model was only 27.5%.

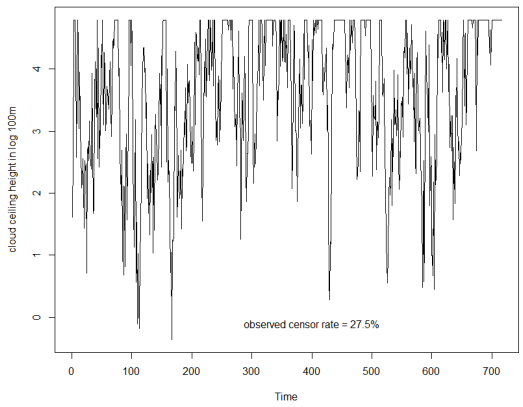

Figure 1.3: Time series plot of simulated censored AR(2) series for the model that was fit to the cloud ceiling time series. The observed censor rate was 27.5% which was lower than the 41.62% rate in the observed historical time series.

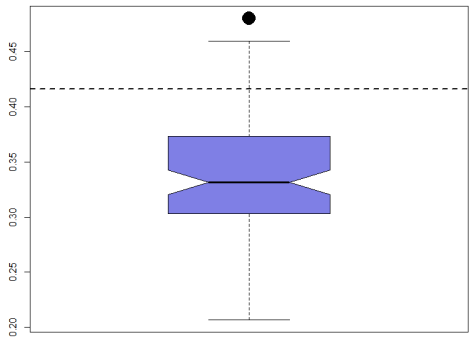

Figure 1.4: Boxplot of the censor rate in 100 simulations of the AR(2) model fitted to the cloud ceiling time series. The dotted horizontal line shows the observed censor rate in the observed data.

EM algorithm

The EM algorithm works, that is under regularity conditions such as for sampling from the exponential family of distributions, due to properties of convex functions including Jensen's inequality and the Kullback-Liebler discrepancy. We will now discuss these concepts and then use them in the derivation of the EM algorithm.

Convex functions

The function $\varphi(x)$ is convex on an interval (a, b) provided that

$$\varphi\left(\lambda x_1 + (1-\lambda)x_2\right) \leq \lambda\varphi\left(x_1\right) + (1-\lambda)\varphi\left(x_2\right) \tag{1.1}$$

for $x_1, x_2 \in (a, b)$ and $0 \leq \lambda \leq 1$. It is strictly convex if equality only holds when $\lambda = 0$ or $\lambda = 1$. Geometrically this means that the chord joining any two points lies above the function as shown in Figure 1.5.

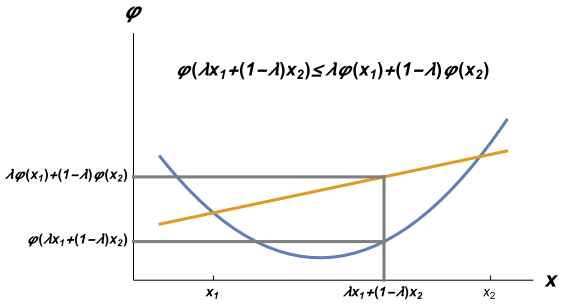

Figure 1.5: A strictly convex function.

When $\varphi(x)$ is twice differentiable, it is strictly convex on $\mathcal{D}$ on (a, b) if and only if $\varphi''(x) > 0$. The slope of the tangent is always increasing. The diagram above shows a parabola that is concave upwards. Other common examples of convex functions are exponential on $(-\infty, \infty)$ and the negative of the logarithm on $(0, \infty)$.

The region above a convex function is always a convex set, that is, the line segment connecting any two points in the region is in the region.

Jensen's inequality

Consider a distribution with two mass points $\Pr\{X = x_i\} = p_i, i = 1, 2$. Let $\varphi(x)$ be a convex function. By definition, $\mathrm{E}\varphi(X) = \varphi\left(x_1\right) p_1 + \varphi\left(x_2\right) p_2$ and $\varphi(\mathrm{E}X) = \varphi\left(p_1 x_1 + p_2 x_2\right)$, where E denotes mathematica expectation.

Then Jensen's inequality follows from the above two results and it states that,

$$\varphi\left(p_1 x_1 + p_2 x_2\right) \leq \varphi\left(x_1\right) p_1 + \varphi\left(x_2\right) p_2 \tag{1.2}$$

This proof can be extended to n points using mathematical induction and then it can be extended to the continuous case using continuity arguments. In general, for any random variable X, if φ is any convex function for which $\mathrm{E}\varphi(X)$ is defined then Jensen's inequality states that,

$$\varphi(\mathrm{E}X) \leq \mathrm{E}\varphi(X) \tag{1.3}$$

Illustrative example with the lognormal distribution. Let X have a lognormal distribution with parameters μ and σ^2. So $Y = \log X$ is normally distributed with mean μ and variance σ^2. Then it can be shown from Jensen's inequality that $\mathrm{E}\log X > \mu$. This follows since $-\log X$ is a strictly convex function over the positive real line, $\mathrm{E}(-\log(X)) > -\log \mathrm{E}X$ from eqn.(1.3) so $\mu < \log \mathrm{E}X$.

This result can be verified using the moment generating function for the normal distribution with parameters μ and σ^2. Let Y be normally distributed with mean μ and variance σ^2 then the moment generating function for Y is given by $M_Y(t) = \mathrm{E}\exp(tY) = \exp\left\{t\mu + \frac{1}{2}t^2\sigma^2\right\}$ hence setting $t = 1$, $\mathrm{E}\{\exp(Y)\} = \mathrm{E}\{X\} = \exp\left\{\mu + \frac{1}{2}\sigma^2\right\} > e^{\mu}$, hence $\mathrm{E}X > e^{\mu}$ or equivalently $\log \mathrm{E}X > \mu$. More generally, if Y is a positive random variable, $\log \mathrm{E}Y \leq \mathrm{E}\log Y$ and equality holds only in the constant random variable case.

Kullback-Liebler discrepancy

Jensen's inequality may be used to establish the non-negativity of the Kullback-Liebler discrepancy. For any two probability density functions $f(x)$ and $g(x)$ defined on $\mathbb{R}$, the Kullback-Liebler discrepancy is defined by

$$K(g, f) = \int_{-\infty}^{\infty} \log \frac{f(x)}{g(x)}\, f(x)dx = \mathrm{E}_f \log \frac{f(x)}{g(x)} \tag{1.4}$$

Proof

$$K(g, f) = \int \log\left(\frac{f(x)}{g(x)}\right) f(x)dx = -\int \log\left(\frac{g(x)}{f(x)}\right) f(x)dx \geq -\log \int \left(\frac{g(x)}{f(x)}\right) f(x)dx = 0 \tag{1.5}$$

This establishes that $K(g, f) \geq 0$.

Incomplete data models

By incomplete data we mean data that is possibly missing or censored. Incomplete data models are characterized by,

$$g(x|\theta) = \int_Z f(x, z|\theta)dz \tag{1.6}$$

where x is the observed data and z is the latent data and $g()$ and $f()$ are the probability density functions. As in [Robert and Casella, 2004] we use the notation $g(x|\theta)$ to mean the density function of x given θ and similarly with others distribution functions.

Incomplete data problems also arise in other areas such as mixture models and stochastic volatility model in financial time series [Robert and Casella, 2004]

Censored data likelihood

Observed $Y_1, \ldots, Y_n$ from IID with pdf $f(y; \theta)$. Assume $y_1, \ldots, y_m$ are fully observed and $y_{m+1} = \ldots = y_n = c$ are left-censored. The likelihood function is

$$L(\theta|y) = F(c)^{n-m} \prod_{i=1}^{m} f\left(y_i; \theta\right). \tag{1.7}$$

Let $z_1, \ldots, z_n$ be latent process with $z_i = y_i$, $i = 1, \ldots, m$ and $z_{m+1}, \ldots, z_n$ are the observed uncensored values. Then the complete likelihood function if we know the latent process values as well as the observed values may be written,

$$L^{(c)}(\theta|z) = \prod_{i=1}^{n} f(z_i;\theta) = \prod_{i=1}^{m} f(y_i;\theta) \prod_{i=m+1}^{n} f(z_i;\theta) \tag{1.8}$$

Then eqn. (1.7) may be written,

$$L(\theta|y) = \mathrm{E}\left\{L^{(c)}(\theta|Z)\right\} = \int_Z L^{(c)}(\theta|z) f(z\,|y_1\,, \ldots, y_m;\theta)\, dz. \tag{1.9}$$

Note that eqn. (1.7) is of the same form as eqn. (1.9).

The EM Algorithm

In the following sections for extra precision we will use boldface to denote vectors. In later sections the distinction between vector and scalars is evident from context so the use of boldface for vectors is curtailed.

Expectation equation

We suppose $X_i, i = 1, \ldots, m$ are independent and identically distributed (IID)

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\mathrm{argmax}} \Pi_{i=1}^{m} g(x_i|\boldsymbol{\theta}) \tag{1.10}$$

and we can write the log-likelihood,

$$\log L(\boldsymbol{\theta}|\boldsymbol{x}) = \log g(\boldsymbol{x}|\boldsymbol{\theta}) \tag{1.11}$$

where $\boldsymbol{x} = (x_1, \ldots, x_m)$. In practice, $\log L(\boldsymbol{\theta}|\boldsymbol{x})$ may be difficult or cumbersome to evaluate.

Suppose that if we augment the data with $\boldsymbol{z} = (z_{m+1}, \ldots, z_n)'$ where (X, Z) have PDF $f(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$ where $f(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$ is easy to evaluate. The complete data likelihood,

$$\log L^{(c)}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{z}) = \log(f(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})). \tag{1.12}$$

The marginal PDF for $\boldsymbol{z}$ is given by

$$k(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{x}) = \frac{f(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})}{g(\boldsymbol{x}|\boldsymbol{\theta})} \tag{1.13}$$

hence,

$$g(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{f(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})}{k(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{x})} \tag{1.14}$$

Taking logs,

$$\log g(\boldsymbol{x}|\boldsymbol{\theta}) = \log f(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) - \log k(\boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{x}) \tag{1.15}$$

For any value of $\boldsymbol{\theta}_0$ we can write,

$$\mathrm{E}_{\boldsymbol{\theta_0}} \log L(\boldsymbol{\theta}|\boldsymbol{x}) = \mathrm{E}_{\boldsymbol{\theta_0}} \log L^{(c)}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\theta}_0) - \mathrm{E}_{\theta_0} \log k(\boldsymbol{Z}|\boldsymbol{x}, \boldsymbol{\theta}_0) . \tag{1.16}$$

where E_{θ_0} is expectation for $\boldsymbol{Z}$ with respect to the distribution $k(\boldsymbol{z}|\theta_0, \boldsymbol{x})$. Since the first term does not depend on $\boldsymbol{Z}$,

$$\log L(\boldsymbol{\theta}|\boldsymbol{x}) = \mathrm{E}_{\theta_0} \log L^{(c)}(\theta|\boldsymbol{x}, \boldsymbol{Z}) - \mathrm{E}_{\boldsymbol{\theta_0}} \log k(\boldsymbol{Z}|\boldsymbol{x}, \boldsymbol{\theta}_0) . \tag{1.17}$$

Assume, as usual, that we can interchange expectation with respect to $\boldsymbol{Z}$, and differentiation with respect to θ_0,

$$\partial_{\boldsymbol{\theta}_0} \mathrm{E}_{\boldsymbol{\theta_0}} \log k(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}_0) = \mathrm{E}_{\theta_0} \partial_{\theta_0} \log k(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}_0) = 0 \tag{1.18}$$

where we have used the result that the expected value of the score function is zero [Casella and Berger, 2002, eq. 7.3.8]. Eqn. (1.18) shows that $\partial_{\boldsymbol{\theta}_0} \mathrm{E}_{\boldsymbol{\theta_0}} \log k(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}_0)$ is independent of $\boldsymbol{\theta}$ and so we can concentrate of maximizing $\mathrm{E} \log L^{(c)}(\theta|\boldsymbol{x}, \boldsymbol{Z})$

Simple illustration

This example is provided just to show in an over-simplified setting how the concepts can be apply. Suppose $X_i, i = 1, \ldots, m$ and $Z_i, i = m+1, \ldots, n$ are IID normal with mean θ and unit variance.

Dropping the constant terms involving 2π,

$$\log L(\theta|\boldsymbol{x}) = \sum_{i=1}^{m} \left(-\frac{1}{2} (x_i - \theta)^2 \right)$$

$$\log L^{(c)}(\theta|\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{m} \left(-\frac{1}{2} (x_i - \theta)^2 \right) + \sum_{i=m+1}^{n} \left(-\frac{1}{2} (Z_i - \theta)^2 \right)$$

$$\log k(\boldsymbol{z}|\boldsymbol{x}, \theta) = \sum_{i=m+1}^{n} \left(-\frac{1}{2} (Z_i - \theta)^2 \right)$$

First note that,

$$\begin{aligned} \mathrm{E}_{\theta_0} \left\{ (Z - \theta)^2 \right\} &= \mathrm{E} \left\{ Z^2 - 2\theta Z + \theta^2 \right\} \\ &= \theta_0^2 + 1 - 2\theta\theta_0 + \theta^2 \\ &= (\theta_0 - \theta)^2 + 1. \end{aligned}$$

So we have,

$$\mathrm{E}_{\theta_0} \left(\sum_{i=m+1}^{n} (-\frac{1}{2} (Z_i - \theta)^2) \right) = -\frac{1}{2}(n-m) \left(1 + (\theta_0 - \theta)^2 \right)$$

Hence,

$$\mathrm{E}_{\theta_0} \log L^{(c)}(\theta|\boldsymbol{x},\boldsymbol{Z}) = \sum_{i=1}^{m}\left(-\frac{1}{2}(x_i-\theta)^2\right) - \frac{1}{2}(n-m)\left(1+(\theta_0-\theta)^2\right)$$

and

$$\mathrm{E}_{\boldsymbol{\theta}_0} \log k(\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}) = -\frac{1}{2}(n-m)\left(1+(\theta_0-\theta)^2\right).$$

So eqn. (1.17) is verified as correct.

EM equation

To maximize $\log L(\boldsymbol{\theta}|\boldsymbol{x})$ we can work with the expected log-likelihood function,

$$Q\left(\boldsymbol{\theta}\,|\boldsymbol{\theta}_0\,, x\right) = \mathrm{E}_{\boldsymbol{\theta}_0} \log L^{(c)}\left(\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{z},\boldsymbol{\theta}_0\right). \tag{1.19}$$

The iterative algorithm starts with an initial estimate $\hat{\boldsymbol{\theta}}_0$ and then obtains improved estimates, $\hat{\boldsymbol{\theta}}_{(1)}, \hat{\boldsymbol{\theta}}_{(2)}, \ldots$. The improved estimates are obtained using,

$$\hat{\boldsymbol{\theta}}_{(j+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} Q(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}_{(j)}, \boldsymbol{x}). \tag{1.20}$$

There are two basic steps in the algorithm.
Step 1. Expectation: compute the expected value,

$$\mathrm{E}_{\boldsymbol{\theta}_{(j)}} \log L^{(c)}(\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{z}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{(j)}, \boldsymbol{x}). \tag{1.21}$$

Step 2. Maximization: use eqn. (1.20).

Fundamental theorem for the EM algorithm

The sequence $\hat{\boldsymbol{\theta}}_{(j)}$, $j = 0, 1, 2, \ldots$ defined by the EM algorithm satisfies

$$\log L(\hat{\boldsymbol{\theta}}_{(j+1)}|\boldsymbol{x}) \geq \log L(\hat{\boldsymbol{\theta}}_{(j)}|\boldsymbol{x}) \tag{1.22}$$

with equality holding if and only if

$$Q(\hat{\theta}_{(j+1)}|\hat{\boldsymbol{\theta}}_{(j)}, x) = Q(\hat{\theta}_{(j)}|\hat{\boldsymbol{\theta}}_{(j)}, x) \tag{1.23}$$

Proof

Eqn. (1.17) may be written,

$$\log L(\boldsymbol{\theta}|\boldsymbol{x}) = Q(\theta|\hat{\boldsymbol{\theta}}_{(j)}, \boldsymbol{x}) - \mathrm{E}_{\boldsymbol{\theta}_{(j)}} \log k(\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}_{(j)}). \tag{1.24}$$

Hence we can write,

$$\log L(\hat{\boldsymbol{\theta}}_{(j+1)}|\boldsymbol{x}) = Q(\hat{\theta}_{(j+1)}|\hat{\boldsymbol{\theta}}_{(j)}, \boldsymbol{x}) - \mathrm{E}_{\boldsymbol{\theta}_{(j)}} \log k(\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}_{(j)}). \tag{1.25}$$

and

$$\log L(\hat{\boldsymbol{\theta}}_{(j)}|\boldsymbol{x}) = Q(\hat{\boldsymbol{\theta}}_{(j)}|\hat{\boldsymbol{\theta}}_{(j)}, \boldsymbol{x}) - \mathrm{E}_{\boldsymbol{\theta}_{(j)}} \log k(\boldsymbol{Z}|\boldsymbol{x}, \boldsymbol{\theta}_{(j)}). \tag{1.26}$$

So we obtain,

$$\log L(\hat{\theta}_{(j+1)}|\boldsymbol{x}) - \log L(\hat{\theta}_{(j)}|\boldsymbol{x}) = Q(\hat{\theta}_{(j+1)}|\hat{\theta}_{(j)}, \boldsymbol{x}) - Q(\hat{\theta}_{(j)}|\hat{\theta}_{(j)}, \boldsymbol{x}) - \mathrm{E}_{\theta_{(j)}} \log k(\boldsymbol{Z}|\boldsymbol{x}, \theta_{(j+1)}) + \mathrm{E}_{\theta_{(j)}} \log k(\boldsymbol{Z}|\boldsymbol{x}, \theta_{(j)}).$$

Since by definition $Q(\hat{\boldsymbol{\theta}}_{(j+1)}|\hat{\boldsymbol{\theta}}_{(j)}, \boldsymbol{x}) - Q(\hat{\theta}_{(j)}|\hat{\theta}_{(j)}, \boldsymbol{x}) \geq 0$ we see that eqn. (1.22) holds if we have

$$\mathrm{E}_{\boldsymbol{\theta}_{(j)}} \log k(\boldsymbol{Z}|\boldsymbol{x}, \boldsymbol{\theta}_{(j+1)}) \leq \mathrm{E}_{\boldsymbol{\theta}_{(j)}} \log k(\boldsymbol{Z}|\boldsymbol{x}, \boldsymbol{\theta}_{(j)}). \tag{1.27}$$

Re-arranging the terms eqn. (1.27) is equivalent to,

$$\mathrm{E}_{\theta_{(j)}} \log \frac{k(\boldsymbol{Z}|\boldsymbol{x}, \boldsymbol{\theta}_{(j+1)})}{k(\boldsymbol{Z}|\boldsymbol{x}, \boldsymbol{\theta}_{(j)})} \geq 0. \tag{1.28}$$

The inequality in eqn. (1.28) follows from the non-negative definiteness property of the Kullback-Liebler information.

Discussion

In cases where the likelihood function is well behaved, such as for models assuming a member of the exponential family of distributions, the likelihood function has a single unique maximum and in these cases the EM algorithm is guaranteed to converge to the global MLE [McLachlan and Krishnan, 2007, §3.4].

The Fundamental Theorem of EM only guarantees that the likelihood does not decrease. Further conditions are needed to establish that it converges to a stationary point of the likelihood equation $\partial_\theta \log L(\theta|x) = 0$ [Boyles, 1983, Wu, 1983].

This stationary point may be a minimum, maximum or saddlepoint. In some situations it is necessary to experiment with different initial values. A simple example of a non-regular case is provided by the Cauchy distribution since the likelihood function for the location parameter may be multimodal. The method of simulated annealing has been used in such difficult situations to try to find the global MLE [Finch et al., 1989, Robert and Casella, 2004]. Other methods such as genetic optimization or MCMC may also be useful in such cases.

Stochastic EM algorithm

It may be that the expected value $\mathrm{E}_{\boldsymbol{\theta}_{(j)}} \log L^{(c)}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{z}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}_{(j)}, \boldsymbol{x})$ is too difficult to compute analytically but an approximation to it can be obtained by simulation. We need to generate $\boldsymbol{z}$ by drawing repeatedly from the distribution $k(\boldsymbol{z}|x, \boldsymbol{\theta}_{(j)})$. The stochastic EM algorithm comprises the following steps:

Step 1: Select M large enough. Set $\boldsymbol{\theta}_{(0)}$ to an initial parameter estimate and set $j = 0$.

Step 2: Generate a random sample $\boldsymbol{z}^{(i)}, i = 1, \ldots, M$ and compute

$$\bar{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}_{(j)}, x) = \frac{1}{M} \sum_{i=1}^{M} \log L^{(c)}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{z}^{(i)}, \boldsymbol{\theta}_0). \tag{1.29}$$

Step 3: Compute update estimate

$$\hat{\boldsymbol{\theta}}_{(j+1)} = \underset{\boldsymbol{\theta}}{\text{argmax}} \bar{Q}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}_{(j)}, \boldsymbol{x}). \tag{1.30}$$

As $M \to \infty$, the estimates of $\hat{\boldsymbol{\theta}}_{(j+1)}$ will stochastically converge to the true MLE, $\hat{\boldsymbol{\theta}}$. This algorithm often referred to as the Monte Carlo EM or MCEM algorithm.

Example comparing the EM and MCEM algorithms

When the complete sample PDF, $f(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$, is a member of the exponential family, the evaluation of eqn. (1.20) or eqn. (1.30) can be simplified [Robert and Casella, 2004, p. 191]. Following Robert and Casella [2004] we consider estimation of the mean parameter θ in a normal distribution with known variance equal to one subject to censoring. We will assume left censoring with the censor point, c, determined by $c = \Phi^{-1}(r)$, where r is a specified censor rate. We suppose that $y_1, \ldots, y_m$ are fully observed and that there are $n - m$ left-censored values reported. Let $\bar{y}$ be the sample mean of $y_1, \ldots, y_m$ and we may initial set $\hat{\theta}_{(0)} = (m\bar{y} + (n-m)c)/n$ and the EM algorithm reduces to iterating the equation,

$$\hat{\theta}_{(j+1)} = (m/n)\bar{y} + (n-m)/n \mathrm{E}_{\hat{\theta}_{(j)},c}\{Z\}. \tag{1.31}$$

where $\mathrm{E}_{\hat{\theta}_{(j)},c}\{Z\}$ is the expected value of a right-truncated normal random variable with mean $\hat{\theta}_{(j)}$ and truncation point c. If we use right censoring, then we take expectations for a left-truncated normal variable. These iterations quickly converge to the true MLE, $\hat{\theta}$.

For the MCEM version, we simulate $z_{m+1}^{(i)}, \ldots, z_n^{(i)}, i = 1, \ldots, M$ from the right truncated normal distribution on $(-\infty, c)$ and then compute the mean to obtain $\hat{\theta}^{(j)}$, as summarized in eqn. (1.32) below,

$$\hat{\theta}^{(j)} = \bar{z}^{(i)}. \tag{1.32}$$

Then we compute the updated estimate,

$$\hat{\theta}_{(j+1)} = (m/n)\bar{y} + (n-m)/n\hat{\theta}_{(j)}. \tag{1.33}$$

Figures 1.6 and 1.7 compare the EM and MCEM algorithms using $M = 100$ and $M = 1$ for the number of Monte-Carlo simulations. In these simulations the sample size was $n = 50$ with a 50% left-censoring rate. We used $N = 25$ EM iterations for both the regular EM and the Monte-Carlo EM. The blue dots, which are connected by line segments, show the EM iterations using eqn. (1.31) while the red dots show the MCEM iterations using eqn. (1.32)and(1.33) with $M = 100$ simulations at each iteration. In Figure 1.6 the convergence is much slower in the Monte-Carlo since it only converges stochastically as both $M \to \infty$ and $N \to \infty$. Observe also that the convergence in the Monte-Carlo case is non-monotonic whereas the Fundamental Theorem of EM guarantees monotonic convergence in the deterministic case.

Figure 1.7 repeats the simulation using the same data values as in Figure 1.6 for the observed time series but using only $M = 1$. As can be seen the procedure does not converge and is useless. McLeod and Mohammad [2013a] provide an interactive demonstration to explore convergence of the MCEM algorithm other various scenarios.

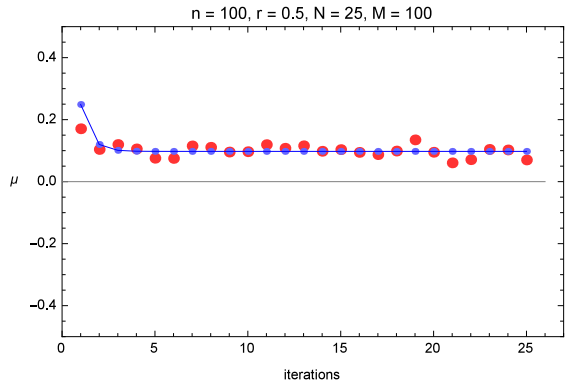

Figure 1.6: Convergence of EM and MCEM after 25 iterations with $M = 100$ for sample size $n = 50$ with censor rate $r = 0.5$.

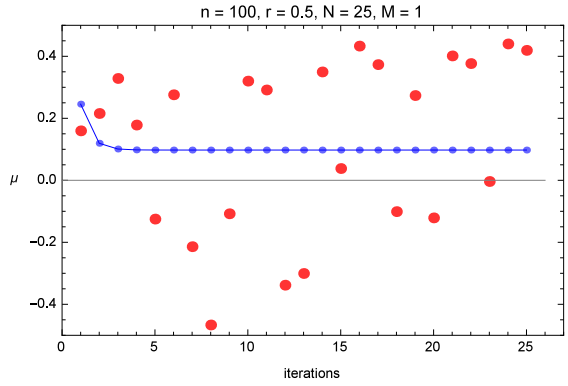

Figure 1.7: Comparing EM and MCEM after 25 iterations with $M = 1$ for sample size $n = 50$ with censor rate $r = 0.5$.

Park et al. [2007] propose a stochastic EM algorithm for fitting ARMA models to censored time series. However in their data augmentation, they use only $M = 1$. Hence the algorithm, as described in their paper, can not be expected to give sensible results. Increasing M in this algorithm would make the computations involved very laborious. At each iteration, the Gibbs algorithm of Robert [1995] for simulating from a truncated multivariate distribution is used and this Gibbs algorithm, like most MCMC methods, does not have any obvious stopping rule. Instead in Chapter 3 we propose a computational efficient quasi-EM method. In the early days of data augmentation some authors used only one simulation but informed researchers no longer do to this.

Oakes theorem

For completeness a derivation following the method described by Oakes [1999] is provided with some additional explanations. We drop the bold face notation used in the previous sections for vectors since it is clear when a variable is scalar or vector from the context. Also following Oakes [1999] we shall use the notation θ and θ' to denote the values of the parameters in the EM algorithm with θ being the current estimated parameter and θ' the parameter to be optimized over and matrix transpose will be denoted by a superscript T. Our derivation is essentially the same as in Oakes [1999] but a few more details and explanations are given.

In Chapter 2, we use Oakes theorem to derive a new result for the observed and expected information in random samples from a normal population.

Oakes [1999] provides an convenient algorithm for computing the information matrix for the parameters when the EM algorithm is used. Other methods have been discussed but Oakes [1999] provides a simple expression for the Hessian in terms of the objective function $Q(\theta'|\theta)$ used in the EM algorithm. We will use Oakes theorem to derive a new expression for the observed and expected information matrix in censored normal samples in Chapter 2.

Oakes theorem states that the Hessian matrix is given by,

$$\frac{\partial^2 L(\theta, y)}{\partial^2 \theta} = \left\{ \frac{\partial^2 Q(\theta'|\theta)}{\partial \theta'^2} + \frac{\partial^2 Q(\theta'|\theta)}{\partial \theta' \partial \theta} \right\}_{\theta'=\theta} \tag{1.34}$$

and so the information matrix is,

$$\mathcal{I}(\theta) = -\left\{ \frac{\partial^2 Q(\theta'|\theta)}{\partial \theta'^2} + \frac{\partial^2 Q(\theta'|\theta)}{\partial \theta' \partial \theta} \right\}_{\theta'=\theta} \tag{1.35}$$

Remark 1: The covariance matrix of the MLE estimate $\hat{\theta}$ is $\mathcal{I}^{-1}(\theta)$ which in practice is estimated by $\mathcal{I}^{-1}\left(\hat{\theta}\right)$.

Remark 2: $\mathcal{I}(\theta)$ is the observed information. For models in the exponential family, Efron and Hinkley [1978] showed that the observed information matrix provides more accurate estimates of the standard errors than the expected information matrix $\mathrm{E}\{\mathcal{I}(\theta)\}$.

Derivation

$$L(\theta', y) = Q(\theta'|\theta) - \mathrm{E}_{X|Y,\theta} \log k(x|y;\theta'). \tag{1.36}$$

As in the usual regularity conditions for MLE we assume that the expectation and differentiation can be interchanged, so that we can use the fact that the expected value of the score function is zero,

$$\mathrm{E}_{X|Y,\theta}\frac{\partial \log k(x|y,\theta)}{\partial \theta} = 0 \tag{1.37}$$

For the MLE, the Hessian is equal to the negative of the product of the score function,

$$\mathrm{E}_{X|Y,\theta}\frac{\partial^2 \log k(x|y,\theta)}{\partial \theta^2} = -\mathrm{E}_{X|Y,\theta}\left(\frac{\partial \log k(x|y,\theta)}{\partial \theta}\right)\left(\frac{\partial \log k(x|y,\theta)}{\partial \theta}\right)^T \tag{1.38}$$

Taking first derivatives in eqn. (1.36),

$$\frac{\partial L(\theta', y)}{\partial \theta'} = \frac{\partial Q(\theta'|\theta)}{\partial \theta'} - \mathrm{E}_{X|Y,\theta}\frac{\partial \log k(x|y,\theta)}{\partial \theta'} \tag{1.39}$$

From eqn. (1.37), the last term vanishes when $\theta' = \theta$ and hence the score function for the observed data,

$$\frac{\partial L(\theta', y)}{\partial \theta} = \left(\frac{\partial Q(\theta'|\theta)}{\partial \theta'}\right)_{\theta'=\theta} \tag{1.40}$$

Differentiating in eqn. (1.39) with respect to θ'

$$\frac{\partial^2 L(\theta', y)}{\partial \theta'^2} = \frac{\partial^2 Q(\theta'|\theta)}{\partial \theta'^2} - \mathrm{E}_{X|Y,\theta}\frac{\partial^2 \log k(x|y,\theta)}{\partial \theta'^2} \tag{1.41}$$

Noting that,

$$\frac{\partial^2 L(\theta', y)}{\partial \theta' \partial \theta} = 0 \tag{1.42}$$

Hence differentiating, in eqn. (1.39) with respect to θ

$$0 = \frac{\partial^2 Q(\theta'|\theta)}{\partial \theta' \partial \theta} - \mathrm{E}_{X|Y,\theta}\left(\frac{\partial \log k(x|y,\theta)}{\partial \theta}\right)\left(\frac{\partial \log k(x|y,\theta)}{\partial \theta}\right)^T \tag{1.43}$$

Adding eqns. (1.41) and (1.43) and then using (1.42), eqn. (1.35) is obtained.

Standard errors using the jackknife

In Chapter 2, the exact observed and expected Fisher information matrices are derived for censored normal samples. By inverting these matrices, estimates of the covariance matrix of the maximum likelihood estimates (MLE) may be obtained. However these estimates are based on the standard asymptotic theory of MLE [Knight, 2000, §5.4] and so its applicability in small samples may not hold or the resulting estimates of the standard errors may not be robust against model mis-specification such as non-normality or non-constant variance. The bootstrap and the Tukey jackknife provides a simpler approaches that have greater robustness and small-sample validity [Davison and Hinkley, 1997]. In Chapter 2, we compare estimates

of the standard errors obtained using the Fisher information matrix method with the more computational intensive method using the jackknife.

Miller (1964) provides a review of the jackknife method and brief overview of its history. The original idea was in a paper by Quenouille in 1949 who proposed a method for removing the bias in the estimation of serial correlation and it was later extended by Tukey in 1956. Tukey introduced the terminology jackknife to indicate that this was a method of wide applicability. In modern practice, the bootstrap provides a similar computationally intensive method that is more generally applicable to statistical inference problems. In Chapter 2 we use the jackknife to compare our asymptotic standard deviations with the jackknife estimates while in Chapter 3 the bootstrap is used.

Let $X = (X_1, \ldots, X_n)$ be a sample and let $g_n(X)$ be an estimator of θ. Tukey defined the ith pseudo-value of $g_n(X)$ is defined to be

$$u_i(X) = ng_n(X) - (n-1)g_{n-1}\left(X_{[i]}\right) \tag{1.44}$$

where $X_{[i]}$ is X with the ith value removed. Then $u_i(X)$ is a bias corrected version of $g_n(X)$. using the ith observation. The jackknife treats $u_i = u_i(X), i = 1, \ldots, n$ as n independent estimators of θ. Let $\bar{u}$ and s_u^2 denote the usual sample mean and variance of $u_i, i = 1, \ldots, n$, that is,

$$\bar{u} = n^{-1} \sum_{i=1}^{n} u_i \tag{1.45}$$

and

$$s_u^2 = \frac{1}{n-1} \sum_{i=1}^{n} (u_i - \bar{u})^2. \tag{1.46}$$

Then the bias-corrected estimate of θ is,

$$\hat{\theta}^{(J)} = \bar{u} \tag{1.47}$$

and the estimated standard error for both $\hat{\theta}$ and $\hat{\theta}^{(J)}$ is

$$\text{est.sd}\left(\hat{\theta}\right) = s_u^2 \Big/ n. \tag{1.48}$$

The jackknife 95% confidence interval is

$$\bar{u} \pm 1.96 s_u / \sqrt{n}. \tag{1.49}$$

Tukey suggested replacing 1.96 in eqn. (1.49) by the upper 2.5% point from a t-distribution on $n-1$ degrees of freedom and this may be useful when n is not large. In general using the jackknife estimate for θ removes the order $1/n$ term in the expansion,

$$\mathrm{E}\left\{g_n(X)\right\} = \theta + a_1/n + O\left(1 \Big/ n^2\right), \tag{1.50}$$

Conditional multivariate normal distribution

Let Z be a vector of length $n_1 + n_2$ and let $Z = \left(Z_1', Z_2'\right)$ be a partitioned vector with Z_1 and Z_2 of lengths n_1 and n_2. Suppose that Z is multivariate normal with mean vector $\mu = \left(\mu_1', \mu_2'\right)$ and covariance matrix

$$\Omega = \begin{pmatrix} \Omega_{1,1} & \Omega_{1,2} \\ \Omega_{2,1} & \Omega_{2,2} \end{pmatrix} \tag{1.51}$$

where $\Omega_{2,1} = \Omega_{1,2}'$. Then the conditional distribution of Z_2 given $Z_1 = z_1$ is normal with mean

$$\mu_{2|1} = \mu_2 + \Omega_{2,1}\Omega_{1,1}^{-1}(z_1 - \mu_1) \tag{1.52}$$

and covariance matrix

$$\Omega_{2|1} = \Omega_{2,2} - \Omega_{2,1}\Omega_{1,1}^{-1}\Omega_{1,2} \tag{1.53}$$

The conditional mean in eqn. (1.52) and conditional covariance matrix in (1.53) do not require the normal assumption and hold under the assumption that the Z, Z_1, and Z_2 have the covariance matrix given in eqn. (1.51).

The provides a direct method for solving the missing value problem in stationary time series. As we shall see the missing value problem is a special case of the censoring problem and this connection will be utilized.

Time series models

Let $z_t, t = 1, 2 \ldots$ be a stationary and ergodic time series with mean μ and autocovariance function $\gamma_k = \mathrm{Cov}(z_t, z_{t-k})$. Given an observed series of length n, the covariance matrix of $(z_1, \ldots, z_n)'$ is given by

$$\Gamma_n = \left(\gamma_{i-j}\right), \tag{1.54}$$

where the (i, j)-entry in the $n \times n$ matrix is indicated. The general linear process (GLP) model may be defined by,

$$z_t = \mu + a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \ldots \tag{1.55}$$

where $a_t, t = 1, 2, \ldots$ is a sequence of independent normal random variables with mean zero, variance σ_a^2 and $\sum_k \psi_k^2 < \infty$.

For many parametric linear time series, $\psi_k, k = 0, 1, \ldots$ are functions of a p-dimensional parameter vector $\beta \in \mathbb{R}^p$. An important example is the stationary and invertible ARMA(p, q) model,

$$z_t - \mu = \phi_1(z_{t-1} - \mu) + \ldots + \phi_p\left(z_{t-p} - \mu\right) + a_t - \theta_1 a_{t-1} - \ldots - \theta_q a_{t-q} \tag{1.56}$$

where $a_t \sim \mathrm{IID}\left(0, \sigma_a^2\right)$. In operator notation, $\phi(B)(z_t - \mu) = \theta(B)a_t$, where $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ and $\theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q$ where B is the backshift operator on t. This model can also

be written, $\phi(B)z_t = \zeta + \theta(B)a_t$, where ζ is the intercept parameter is $\zeta = \phi(1)\mu$. The stationary and invertible requirement may be stated that all roots of the polynomial equation $\phi(B)\theta(B) = 0$ lie outside the unit circle where B in this equation is a complex variable. More generally we also interest in the stationary and invertible autoregressive fractionally integrated moving average ARFIMA(p, d, q) where the model equation may be written, $\nabla^d\phi(B)z_t = \zeta + \theta(B)a_t$, where $|d| < 1/2$ and $\phi(B)$ and $\theta(B)$ are defined as in the ARMA case.

Some important special case are when $q = 0$, we have the family of autoregressive models denoted by AR(p). Similarly when $p = 0$, we have the moving average process that is denoted by MA(q).

Many software packages follow another convention were the moving average component is written, $\theta(B) = 1 + \theta_1 B + \ldots + \theta_q B^q$ and so we must simply take the negative of the estimated MA-coefficients.

Given data, $z = (z_1, \ldots, z_n)'$, the log-likelihood for this model may be written,

$$\log \mathcal{L}\left(\beta, \mu, \sigma_a^2; z\right) = -0.5 \log\left(\det\left(\Gamma_n\right)\right) - 0.5(z-\mu)'\Gamma_n^{-1}(z-\mu). \tag{1.57}$$

In practice, in most cases, as discussed in McLeod et al. [2007, eqns 10-11], it is convenient to work with the concentrated log-likelihood function,

$$\log \mathcal{L}_c(\beta; z) = -0.5n \log(S(\beta)/n) - 0.5g_n \tag{1.58}$$

where

$$S(\beta) = \sum_{t=1}^{n} \frac{(z_t - \hat{z}_t)^2}{\sigma_t^2} \tag{1.59}$$

where $\hat{z}_t$ and σ_t^2 are the conditional mean and variance of z_t given $z_1, \ldots, z_{t-1}$ and

$$g_n = \sum_{t=1}^{n} \log\left(\sigma_t^2\right). \tag{1.60}$$

The innovation variance estimate is $\hat{\sigma}_a^2 = S\left(\hat{\beta}\right)\Big/ n$.

McLeod et al. [2007] discussed an R package ltsa (McLeod et al. [2012]) for the efficient computation of the log-likelihood function in eqn. (1.58) for a wide class of linear time series. Maximum likelihood estimates may be obtained by using a suitable non-linear optimizer such as provided by the R function optim() or *Mathematica*'s FindMinimum[].

After fitting the model it is important to check the adequacy of the model assumptions. The most important assumption for our models is that the innovation sequence $a_t, t = 1, \ldots, n$ should be approximately normally distributed and statistical independent. Moderate departure for normally are usually not important provided that the distribution is symmetric and the tails are not too heavy. But lack of independence may result in incorrect inferences and sub-optimal forecasts. A basic test for lack of independence is the Box-Ljung portmanteau test,

$$Q_m = n(n+2) \sum_{k=1}^{m} \hat{r}_k^2 \Big/ (n-k) \tag{1.61}$$

where $\hat{r}_k$ denotes the autocorrelation at lag k of the residuals, $\hat{a}_t, t = 1, \ldots, n$ assuming the data series is of length n. Under the assumption of model adequacy, $Q_m \sim \chi^2_{m-p-q}$ and large values of Q_m indicate model inadequacy. In R, a plot is constructed shown the p-value of Q_m for $m = 1, 2, \ldots, M$, where M is a pre-selected maximum lag.

A water quality application

Dr. Abdel El-Shaarwai provided through Environment Canada some a special water quality time series that is of great practical interest. The time series is from Station ON02HA0019 (Fort Erie) on the water quality of the Niagara River. There are more than 500 water quality parameters or variables of interest in this river. The water quality in this river is monitored by a joint U.S./Canada committee. One important toxic variable of great interest is a chemical known as 12-Dichloro which when dissolved in water is measured in units of ng/L. We use a portion of the recent data on this variable that was measured approximately every two weeks over the period from March 1, 2001 to March 22, 2007. This period was chosen because it is the most recent period over which we have a time series of approximately biweekly observations. The time series plot in Figure 1.8 plots the Julian day number defined so that Julian day number 1 corresponds to the date of the first observation (March 1, 2001). The 123 blue points correspond to full observations while the 21 red ones are left-censored. In total there are 144 values shown in the plot. The observed censoring rate is $r = 21/144 = 14\%$. The red line at the bottom indicates the detection level. After March 24, 2005 the detection level for 12 Dichloro dropped from 0.214 to 0.0878. After this change there was only one censored value at Julian day number 1807. Before the change in censoring there were 75 complete observations and 20 censored ones while from March 24, 2005 to the last observation on March 22, 2007 there were 48 complete observations and only one censored observation. It is evident from the time series plot that the data distribution is positively skewed the autocorrelations are not large. Also note change in detection level on March 24, 2005 and the apparent increase after the change. We need to be careful though since it is possible this change could be simply due to autocorrelation effect since there is no prior reason to suspect the change in detection level was related to the apparent increase in toxic level of 12-Dichloro.

This time series is available in our R package cents.

One approach could involve treating these gaps as missing values but due to the weak autocorrelation in the data this approach would not be expected to produce much improvement. So as an approximating we regard the series as successive measurements spaced approximately every two weeks with a few longer spacing. As a first approximation we ignore the censoring and treat the censored values as actual observations. Figure 1.9 shows an estimate of the probability density function using Gaussian kernel with the bandwidth chosen by cross-validation. The plot shows that the data has a skewed distribution similar to a log-normal or gamma.

The sample skewness $g_1 = 2.76$, kurtosis $g_2 = 13.12$ and shape of the distribution are in better agreement with a log-normal distribution than a gamma distribution. The log-normal and gamma distribution are both useful in fitting positive real valued data. The difference between these distributions was explored in the Wolfram Demonstration El-Shaarawi et al. [2013] where it was shown that after fixing the shape parameter in both distributions by constraining the means to be one and the coefficient of variation to be fixed, the skewness and kurtosis

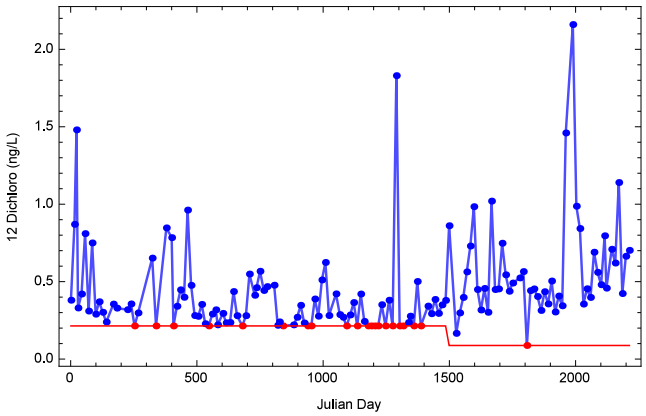

Figure 1.8: Time series plot of 12 Dichloro in Niagara River at Fort Erie.

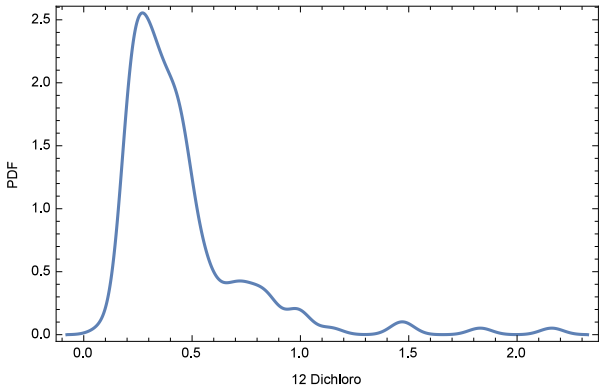

Figure 1.9: The probability density function using Gaussian kernel of 12 Dichloro in Niagara River at Fort Erie.

coefficients are always larger in the log-normal distribution. We will proceed to analyze the logarithms of the 12 Dichloro.

The boxplot shown below shows that log transformed data is more symmetric but there is still some evidence of right skewness. The skewness and kurtosis coefficients have been reduced to $g_1 = 0.69$ and $g_2 = 3.80$ in the logged series.

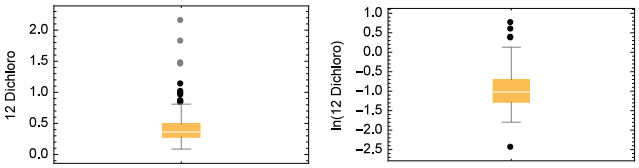

Figure 1.10: Log transformed 12 Dichloro

The autocorrelation plot reveals that the autocorrelations are quite small with, $r_1 = 0.315$ and $r_2 = 0.220$. The red-lines in the plot below indicate 95% benchmark limits estimated using Bartlett's large-lag formula [Box et al., 2008].

$$\text{var}\,(r_k) = \left(1 + \rho_1^2 + \rho_2^2\right) \Big/ n \tag{1.62}$$

The autocorrelation plot, Figure 1.11, reveals that the autocorrelations are quite weak. This plot suggests that AR(1) might not be the best model. Either an MA(2) or ARMA(1,1) would likely produce a better fit.

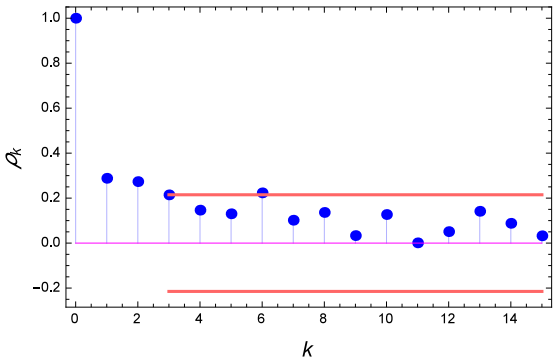

Figure 1.11: Autocorrelation plot of Log transformed 12 Dichloro

Fitting an ARMA model may be useful for short-term prediction and detecting outliers as well as for obtaining an estimate of the mean level. In order to obtain a valid confidence interval it is necessary to take into account the autocorrelation. As a first step in fitting an ARMA we ignore the censoring and treat censored values as fully observed. An approach such as this is often called a naive approach [Helsel, 2011, Ch. 1]. Such naive approaches may lead to incorrect and seriously biased inferences if the censor rate is moderate but with low censor rates a naive approach may be sufficient for practical purposes. The Wolfram Demonstration McLeod and Mohammad [2013b] illustrates the effect of censor rate on the estimation of the mean in normal random samples. Since the detection point is low and the censor rate of about

14% is not high, this approximation seems not unreasonable. The panel below shows the R script used for fitting an AR(1) and ARMA(1,1) model. The diagnostic plots in the Figures 1.13 and 1.12,suggest the AR(1) is borderline adequate while the ARMA(1,1) is completely satisfactory.

The display below shows the fitting the naive ARMA(1,1) model to the logarithms of the 12 Dichloro time series.

```
> require("cents")
> Zdf <- NiagaraToxic
> z <- log(Zdf$toxic)
> iz <- c("o", "L")[1+Zdf$cQ]

> #AR(1)
> ans <- arima(z, order=c(1,0,0))
> ans

Call:
arima(x = z, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.2889    -0.9468
s.e.  0.0798     0.0574

sigma^2 estimated as 0.2408:  log likelihood = -101.86,  aic = 209.73

> #ARMA(1,1)
> ans <- arima(z, order=c(1,0,1))
> ans

Call:
arima(x = z, order = c(1, 0, 1))

Coefficients:
         ar1      ma1  intercept
      0.9152  -0.7380    -0.9043
s.e.  0.0903   0.1553     0.1251

sigma^2 estimated as 0.2241:  log likelihood = -96.83,  aic = 201.66
```

In these diagnostic plots in Figures 1.13 and 1.12, the R function tsdiag() was used for the diagnostic checks. This function produces a plot of the p-values of the Box-Ljung portmanteau diagnostic check vs. lag k, where $k = 1, 2. \ldots, M$, where M is pre-selected. If all p-values are above the 5% limit, we say the fit is satisfactory whereas in there are many p-values less than 5% the fit is clearly not satisfactory. In between these two extremes, we label the fit

borderline. In interpreting this plot it must be borne in mind that the successive p-values are not all statistically independent and are in fact very highly correlated.

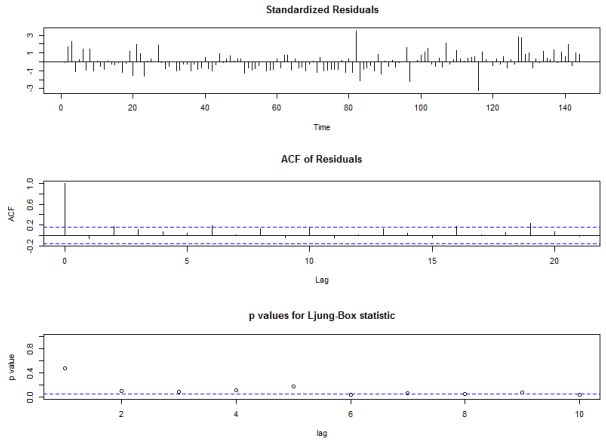

Figure 1.12: Diagnostic plot fitted AR(1) produced by tsdiag().

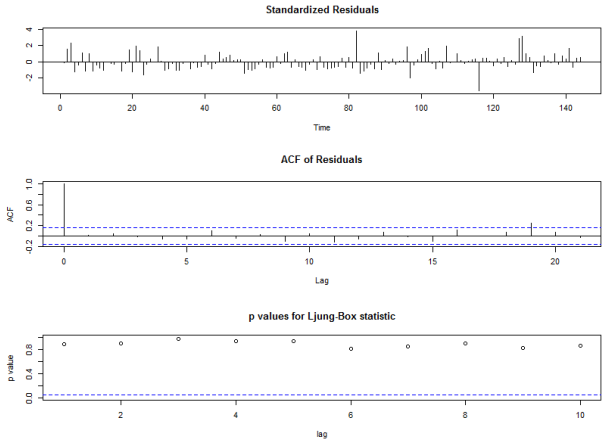

Figure 1.13: Diagnostic plot fitted ARMA(1,1) produced by tsdiag().

For diagnostic checking CENARMA models, we recommend the Monte-Carlo test approach (Lin and McLeod, 2000; Mahdi and McLeod, 2000). A script for the Monte-Carlo test is given in the documentation of the NiagaraToxic variable in the cenarma package. For comparison, the Monte-Carlo diagnostic plots for the and ARMA(1,1) and AR(1) using the Ljung-Box statistic are shown in Figures 1.14 and 1.15 and we conclude that they agree with the asymptotic test results.

From the autocorrelation plot, an ARMA(1,1) was suggested and indeed it was found that this model gave the best fit in terms of AIC as well as being adequate in terms of the portmanteau diagnostic check. The estimated parameters where $\hat{\phi}_1 = 0.9152 \pm 0.0903$, $\hat{\theta}_1 = 0.7380 \pm 0.1553$, $\hat{\mu} = -0.9043 \pm 0.1251$, and $\hat{\sigma}_a^2 = 0.2299$. Note that when fitting with the R function arima(), their ARMA definition uses the negative of our definition, so we have

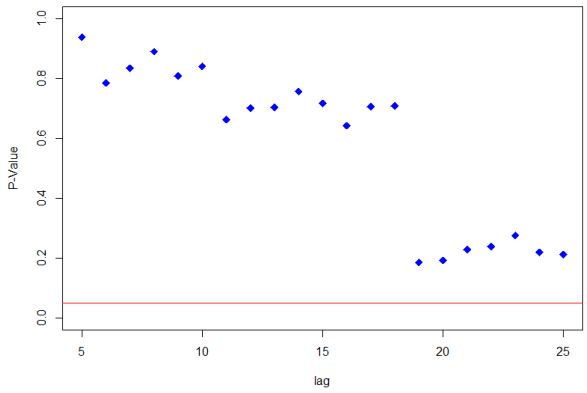

Figure 1.14: Monte-Carlo Ljung-Box test diagnostic plots for fitted ARMA(1,1).

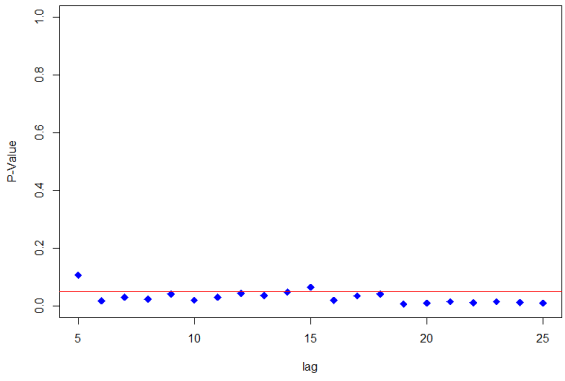

Figure 1.15: Monte-Carlo Ljung-Box test diagnostic plots for fitted AR(1).

adjusted the answer to reflect this. A second point to note that the parameter μ refers to log 12 – Dichloro.

Various other models were examined and their performance is summarized in Table 1.1.

Model	AIC	Portmanteau Diagnostic
ARMA(1,1)	201.66	satisfactory
ARMA(2,0)	205.13	satisfactory
ARMA(0,3)	207.60	satisfactory
ARMA(0,2)	209.48	borderline
ARMA(1,0)	209.73	failed

Table 1.1: Models fit to log 12 Dichloro time series ignoring censoring.

It is interesting that the fitted ARMA(1,1) model implies that although the autocorrelations are small, they decay quite slowly so the series has in a sense a longer memory than the other models in the above table. This series is also well fit by a stationary ARFIMA(0,1,0) model using the software Veenstra and McLeod [2014]. This longer memory is reflected in the theoretical autocorrelation and power spectral density plots for the fitted ARMA(1,1) in Figures 1.16 and 1.17.

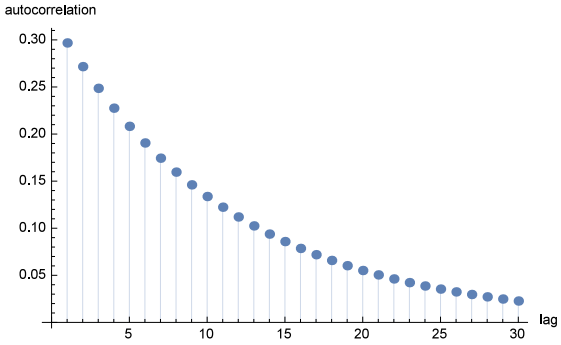

Figure 1.16: Autocorrelation function of the fitted ARMA(1,1) process.

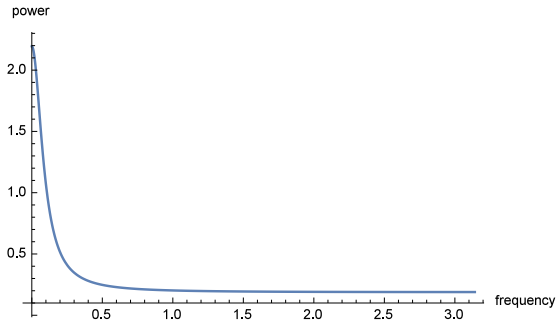

Figure 1.17: Spectral density function of the fitted ARMA(1,1) process.

Chapter 2

Censored normal random samples

Introduction

Before developing an EM algorithm for fitting censored linear time series model we first discuss the simpler case of estimating the parameters μ and σ^2 in random samples from a normal distribution with mean μ and variance σ^2, that is, for random samples from the NID$\left(\mu, \sigma^2\right)$ distribution. This algorithm is used to provide initial estimates of the mean in the EM algorithm that is developed for the linear model case.

A new derivation of the EM algorithm for maximum likelihood estimation (MLE) for left and right censored data with multiple censor points. The main new result in this Chapter is an explicit formula is derived for the expected and observed Fisher information matrix and it is shown that the expected information matrix gives new insight into the statistical behavior of the MLE estimates.

Another new developments is the dynamic normal probability plot of robust estimation and diagnostic checking for censored samples,

The Chapter concludes with an application to the well-known electronic locomotive engines dataset as well as the toxic water quality on 12 Dichloro in the Niagara river discussed in Chapter 1.

Censored sample distribution function and likelihood

Consider the more general left-censored case with latent process $(Z_i, c_i), i = 1, \ldots, n$ where $Z_i, i = 1, \ldots, n$ are independent and identically distributing with probability density function $f(z; \theta)$ and cumulative distribution function $F(z; \theta)$, where $c_i, i = 1, \ldots, n$ are known constants or if random they are assumed to be known and to be statistically independent of $Z_i, i = 1, \ldots, n$. If censoring is not applicable we set $c_i = -\infty$ for those observations for which there is no censor point. Hence, in general, the observed process consists of some fully observed values for which $Z_i > c_i$ and some censored values for which $Z_i \leq c_i$.

Without loss of generality we may re-order the observations so the first m correspond to fully observed values, $Y_i, i = 1, \ldots, m$. Taking into account that number ways the m fully-observed values can be selected the probability density function for the random sample may be written,

$$f_L(y_1,\ldots,y_m;\theta) = \binom{n}{m} \prod_{i=1}^{m} f(y_i;\theta) \prod_{i=m+1}^{n} F(c_i;\theta). \tag{2.1}$$

The corresponding log-likelihood, after dropping the constant term, may be written,

$$\log L(\theta|y,m) = \sum_{i=1}^{m} \log f(y_i;\theta) + \sum_{i=m+1}^{n} \log F(c_i;\theta). \tag{2.2}$$

In the single-left-censored case with detection level c and parameters $\theta = (\mu,\sigma)$,

$$f_L(y_1,\ldots,y_m,|\mu,\sigma,c,m) = \binom{n}{m} F(c;\mu,\sigma)^{n-m} \prod_{i=1}^{m} f(y_i;\mu,\sigma) \tag{2.3}$$

and the log-likelihood function may be written,

$$\log L(\mu,\sigma|y,m) = (n-m)\log F(c;\mu,\sigma) + \sum_{i=1}^{m} \log f(y_i;\mu,\sigma). \tag{2.4}$$

Eqn. (2.2) is equivalent to the expression given by Cohen [1991, eqn. 1.5.3] and Lawless [2003].

If censoring is ignored and we only consider the m fully observed values then $Y_i, i = 1,\ldots,m$ are distributed from a left truncated normal distribution with truncation points $c_i, i = 1,\ldots,m$ and similarly the unobserved latent random variables $Z_{m+1},\ldots,Z_n$ are from a right truncated normal distribution with truncation points $c_{m+1},\ldots,c_n$. In the latent probability model involving $X_1,\ldots,X_n$ with left-censoring at c, the number of complete observations, M, is a random variable and $m = \mathrm{E}\{M\} = n(1-F(c;\mu,\sigma))$. The observed m plays the role of an ancillary statistic and this should be taken into account for statistical inferences on the parameters μ and σ.

If the distribution is symmetric, as in the case of the normal distribution, $f(y_i;\mu,\sigma) = f(-y_i;-\mu,\sigma)$ and $F(c;\mu,\sigma) = 1 - F(-c;-\mu,\sigma)$. Consequently we see that, from a computational viewpoint, the algorithms we develop for the left-censored Gaussian case may be used with right-censoring simply be negating the data and then transforming the estimates back to the original data domain, that is, by negating them again.

In the most general case we allow multiple left and right censor points so $c_i = \left(c_i^-,\ c_i^+\right)$, where c_i^- and c_i^+ denote the left and right censor points respectively. Setting $c_i^+ = \infty$ means that there is no censor point for the ith observation. In practice the most common situation is where there is a single censor point. For left-censoring this means, $c_i = (c,\infty)$, $i = 1,\ldots,n$.

In principle it is straightforward to obtain the maximum likelihood estimates by numerically optimizing the appropriate log likelihood function using a general purpose optimizer such as `FindMaximum[]` in *Mathematica* or `optim()` in R. However we will show that the EM algorithm provides a more computationally efficient approach that is easily implemented. An advantage of the EM algorithm is that convergence is guaranteed as will discussed further later. For estimation of the mean and variance in normal samples the EM algorithm is very fast, so lengthy simulations such as for bootstrapping or jackknifing can be done typically in a second or two. The algorithms discussed in this Chapter have been implemented in *Mathematica* and R.

Maximum likelihood estimation

Left-censored normal random samples

In the first instance we consider singly left-censored data with $c = c_i^-, i = 1, \ldots, n$. Approximate maximum likelihood methods, using tables and charts, for estimating μ and σ^2 in the case of the normal distribution were given by Gupta [1978]. The log-likelihood function can be written,

$$L(\theta|y) = (n-m)\log\Phi(c_z) - (m/2)\log\left(\sigma^2\right) - (1/2)\sum_{i=1}^{m}(y_i-\mu)^2/\sigma^2 \tag{2.5}$$

where Φ is the standard normal CDF and $c_z = (c-\mu)/\sigma$. In computing environments such as *Mathematica* and R, the likelihood function may be maximized numerically using a general purpose built-in optimizer. The maximum likelihood equations are somewhat complex. Taking the first derivatives,

$$\frac{\partial L}{\partial \mu} = (n-m)\phi(c_z)(-1/\sigma)/\Phi(c_z) + \sum_{i=1}^{m}(y_i-\mu)/\sigma^2 \tag{2.6}$$

$$\frac{\partial L}{\partial \sigma^2} = (n-m)\phi(c_z)(-1)(c-\mu)\Big/\sigma^2\,\Phi(c_z) + (m/2)\Big/\sigma^2 + (1/2)\sum_{i=1}^{m}(y_i-\mu)^2/\sigma^4 \tag{2.7}$$

Setting $\partial L/\partial\mu = 0$ we obtain the first MLE equation,

$$\hat{\mu} + \hat{\sigma}(1-n/m)\Psi(c_z) = \bar{y}, \tag{2.8}$$

where $\Psi(z) = \phi(z)/\Phi(z)$. Eqn. (2.8) indicates that the mean of the left-truncated observations, $\bar{y}$, is approximately equal to the true population mean μ plus an upward adjustment that depends on the truncation point. Next setting $\partial L\Big/\partial\sigma^2 = 0$ and simplifying,

$$n\mu(2\bar{y}-\mu) + (n-m)\sigma(c-\mu)\Psi((c-\mu)/\sigma) + m\sigma^2 = n\bar{y} + m\hat{\sigma}_y^2 \tag{2.9}$$

where $\hat{\sigma}_y^2 = m^{-1}\sum_i (y_i-\bar{y})^2$. Cohen (1950, 1991) and Schneider (1986) obtained the maximum likelihood estimations by solving these equations iteratively but this method does not always converge since convergence depends on the initial starting values. Wolynetz (1979) provides an iterative algorithm for solving the likelihood equations $\partial L/\partial\mu = 0$ and $\partial L\Big/\partial\sigma^2 = 0$ in the more general case with multiple left and right censor points. But since his algorithm works directly with the likelihood equations, it is not equivalent to the EM algorithm but rather a variation of the iterative algorithms discussed by Cohen (1950, 1991) and Schneider (1986).

The EM algorithm, discussed next, has the advantage over the iterative methods that it always converges.

Derivation of the EM algorithm for left-censoring with normal distribution

We assume a random sample $y_1, \ldots, y_n$ has been observed from a left-censored normal distribution with known single censor point c and with parameters μ and σ, that denote the mean

and standard deviation in the underlying normal distribution without censoring. We will denote arbitrary values of the parameters by μ' and σ'. Without loss of generality we assume that $y_1, \ldots, y_m$ are not censored and that $y_{m+1} = \ldots = y_n = c$ are left-censored. We introduce latent random variables, $Z_{m+1}, \ldots, Z_n$ which represent the unknown values corresponding to the censored observations. Then $Z_{m+1}, \ldots, Z_n$ are a random sample of size $n-m$ from a right-truncated normal distribution defined on $(-\infty, c)$ and with parameters μ and σ. After dropping constant terms, the complete likelihood can be written,

$$\mathcal{L}^c(\mu', \sigma'|y, z) = \sigma^{-n} \prod_{i=1}^{m} \exp\{-(y_i - \mu)^2/(2\sigma^2)\} \prod_{i=m+1}^{n} \exp\{-(z_i - \mu)^2/(2\sigma^2)\} \tag{2.10}$$

The next step is to compute $Q(\mu', \sigma'|\mu, \sigma, y) = \mathrm{E}_Z\{\log \mathcal{L}^c(\mu', \sigma'|y)\}$. We obtain,

$$Q(\mu', \sigma'|\mu, \sigma, y) = -\frac{n}{2}\log((\sigma')^2) - \frac{1}{2}\sum_{i=1}^{m}(y_i - \mu')^2/(\sigma')^2 - \frac{1}{2}\sum_{i=m+1}^{n}\mathrm{E}_Z\{(Z_i - \mu')^2\}/(\sigma')^2 \tag{2.11}$$

Simplifying, we may re-write this as,

$$Q\left(\mu', \sigma' | \mu, \sigma, y\right) = -\frac{n}{2}\log\left((\sigma')^2\right) - \frac{(\sigma')^{-2}}{2}\left\{\sum_{i=1}^{m}\left(y_i - \mu'\right)^2 + (n-m)\mathrm{E}_Z\left\{\left(Z - \mu'\right)^2\right\}\right\} \tag{2.12}$$

$$\frac{\partial Q}{\partial \mu'} = \sigma^{-2}\left\{\sum_{i=1}^{m}\left(y_i - \mu'\right) + (n-m)\mathrm{E}_{Z(\mu,\sigma,c)}\left\{\left(Z - \mu'\right)\right\}\right\} \tag{2.13}$$

Setting $\partial Q/\partial \mu' = 0$ and solving for $\hat{\mu}$,

$$0 = m\bar{y} - m\hat{\mu} + (n-m)\mathrm{E}_z\{Z\} - (n-m)\hat{\mu}$$

$0 = m\bar{y} + (n-m)\mathrm{E}_z\{Z\} - n\hat{\mu}$. Hence,

$$\hat{\mu} = (m/n)\bar{y} + (n-m)/n\mathrm{E}_z\{Z\}. \tag{2.14}$$

Using *Mathematica*, we obtain the expectation in the right-truncated distribution,

$$\mathrm{E}_Z\{Z|\mu, \sigma, c\} = \left(\mu\mathrm{erf}\left(\frac{c-\mu}{\sqrt{2}\sigma}\right) - \sqrt{\frac{2}{\pi}}\sigma e^{-\frac{(c-\mu)^2}{2\sigma^2}} + \mu\right)/\mathrm{erfc}\left(\frac{\mu - c}{\sqrt{2}\sigma}\right) \tag{2.15}$$

where $\mathrm{erf}(z)$ is the error function defined for $z \geq 0$ as $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$ and $\mathrm{erfc}(z) = 1 - \mathrm{erf}(z)$. In R, it is more convenient to work with the normal distribution so we use the relationships $\mathrm{erf}(z) = 2\Phi\left(\sqrt{2}z\right) - 1$ and $\mathrm{erfc}(z) = 2\left(1 - \Phi\left(\sqrt{2}z\right)\right)$ where $\Phi(z)$ denotes the cumulative distribution function of the standard normal distribution. An equivalent formula for the mean of the truncated normal distribution was derived by Barr and Sherrill [1983]. Next for σ,

$$\frac{\partial Q}{\partial (\sigma')^2} = -\frac{n(\sigma')^{-2}}{2} + \frac{(\sigma')^{-4}}{2}\left\{\sum_{i=1}^{m}(y_i - \mu')^2 + (n-m)\mathrm{E}_{\mathrm{E}_{\mu,\sigma,c}}\left\{(Z-\mu')^2\right\}\right\}. \tag{2.16}$$

Setting $\partial Q \Big/ \partial (\sigma')^2 = 0$ and solving

$$\hat{\sigma}^2 = n^{-1}\left\{\sum_{i=1}^{m}(y_i - \hat{\mu})^2 + (n-m)\mathrm{E}_{\mu,\sigma,c}\left\{(Z-\hat{\mu})^2\right\}\right\} \tag{2.17}$$

where, using *Mathematica*,

$$\mathrm{E}_Z(Z-\mu')^2 = \left(-e^{-\frac{(c-\mu)^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}\sigma(c+\mu-2\mu') + \left(1+\mathrm{erf}\left[\frac{c-\mu}{\sqrt{2}\sigma}\right]\right)\left(\mu^2+\sigma^2-2\mu\mu'+(\mu')^2\right)\right)/\mathrm{erfc}\left[\frac{-c+\mu}{\sqrt{2}\sigma}\right] \tag{2.18}$$

Combining eqns. (2.14) and (2.17), the EM algorithm for computing the MLE can now be written. We start with initial estimates $\hat{\mu}^{(0)}$ and $\left(\hat{\sigma}^2\right)^{(0)}$ that may, in general be obtained by replacing the censored values with the corresponding censor point or perhaps some suitable value or values.

Algorithm CM1. Singly left-censored samples.

Step 1: Initialization: $j \longleftarrow 0$ and `Maxiter` $\longleftarrow 100$

Step 2: $\tilde{\mu}_z \longleftarrow \mathrm{E}_{\hat{\mu}^{(j)},\hat{\sigma}^{(j)},c}\{Z\}$

Step 3: $\hat{\mu}^{(j+1)} \longleftarrow (m/n)\bar{y} + (n-m)/n\tilde{\mu}_z$

Step 4: Compute $\tilde{\sigma}_z^2 \longleftarrow \mathrm{E}_{\hat{\mu}^{(j)},\hat{\sigma}^{(j)},c}\left\{(Z-\mu)^2\right\}$

Step 5: $\left(\hat{\sigma}^2\right)^{(j+1)} \longleftarrow n^{-1}\left\{\sum_{i=1}^{m}(y_i-\hat{\mu})^2 + (n-m)\tilde{\sigma}_z^2\right\}$

Step 6: Test for convergence of the estimates. If they have not converged, $j \longleftarrow j+1$ and repeat Steps 2-5 provided that $j <$ `MaxIter`.

Remarks

1. The EM algorithm can be robustified by replacing the sample mean by some other robust estimator of location. As well a robust estimate for the $\sum_{i=1}^{m}(y_i-\hat{\mu})^2$ could also be introduced. Various state-of-the-art robust estimators for location and scale are available in R Venables and Ripley [2002].

2. The MLE obtained using a general purpose optimization algorithm such as those that are available in FindMaximum in *Mathematica* or optim() in R could also be used but the disadvantage is that convergence may not be guaranteed and these general purpose methods are often much slower.

3. For any symmetric distributions, such as the normal distribution, the analysis of right-censored data may be accomplished by negating the data and apply methods for left-censored analysis. Thus Algorithm CM1 extends directly the to right censored case.

Simulation and other validation checks

As a check on the EM algorithm we compared its performance with direct optimization. For this purpose 1000 simulations were done for a sample of size 50 left censored normal with mean 100, standard deviation 15 and censor rate 25%. The EM algorithm required 2.2 seconds. The simulations were repeated using the same random numbers for the direct MLE method using FindMaximum and this required 21.3 seconds. So there is a ten-fold increase in time. In both cases the algorithms converged on almost identical results for both the estimate of the mean and variance. The boxplots below show the estimates for the means as well as for their difference. It was interesting that the difference was statistically highly significant with a p-value less than 10^{-10} on a two-sample paired t-test even though as is clear from the boxplot of the differences there is no difference of any practical importance. The root-mean-squared error, the average squared difference between the estimate, $\hat{\mu}$, and its true value of $\mu = 100$ was computed for each estimate and difference in relative efficiency in terms of the RMSE was less than 10^{-7} which confirms that there is no practical difference. Similarly the estimates for σ produced by the EM and direct methods were found to be almost identical.

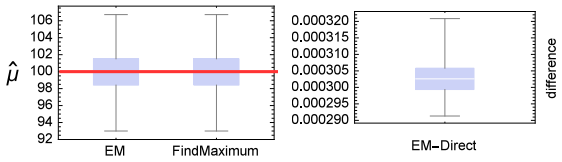

Figure 2.1: Boxplots comparing the estimates of the mean obtained using the EM algorithm and direct numerical optimization using Mathematica's general purpose FindMaximum function.

A version of the EM algorithm for the estimation of Gaussian MLE in normal samples was implemented in Fortran by Wolynetz (1979) and we verified with several examples that our algorithm gave identical results.

Comparing maximum likelihood estimation with crude approximation

A crude estimate for the parameters μ and σ^2 that is often used (Wolynetz, 1979) for an initial estimate in an iterative MLE algorithm is obtained by setting, $y_i = c$, $i = m + 1, \ldots, n$ and then using the sample mean and variance of $y_1, \ldots, y_n$.

Figure 2.2 compares the relative likelihoods of the parameter μ using the crude approximation and the maximum likelihood estimator shown in the dashed red and solid blue curves respectively. The relative likelihood is the likelihood rescaled so its maximum value is 1.0. The three vertical lines provide a visual comparisons of the estimators. The left-most line corresponds to the true parameter value, $\mu = 0$, the next one the maximum likelihood estimate of μ and the right one the crude approximate estimate. We see that approximate has a larger positive bias as might be expected. A *Mathematica* demonstration that allows one to interactively

compare these estimators for varying sample sizes and censoring rates is available (McLeod and Nagham, 2013).

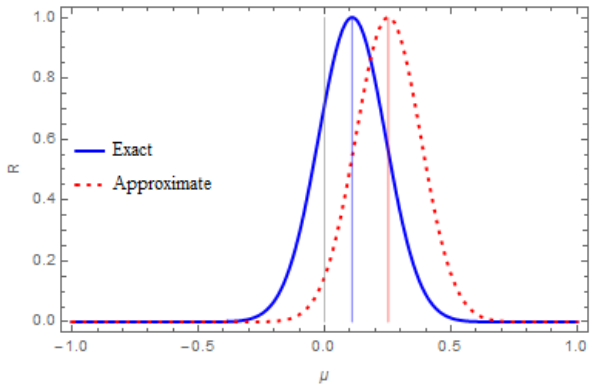

Figure 2.2: The relative likelihood functions using the censored likelihood (solid blue) and the approximation obtained by treating the censored values as observed. In this case the censor rate was about 40 percent so the effect on the bias of the estimate is very strong. As the censor rate decreases the bias will decrease.

Application of the Jackknife to censoring

In our censoring problem we set $Y = (Y_1, \ldots, Y_m)$ and $\theta = (\mu, \sigma)$. Let $\hat{\theta}(Y, n)$ denote the MLE based on a random sample of size n. Let $Y_{[i]}$ be the vector Y with the ith element removed.

Set

$$\hat{\theta}_i = \begin{cases} \hat{\theta}(Y_{[i]}, n-1) & \text{if } i = 1, \ldots, m; \\ \hat{\theta}(Y, n-1) & \text{if } i = m+1, \ldots, n. \end{cases} \tag{2.19}$$

Then the pseudo-values are,

$$u_i = n\hat{\theta}(Y, m) - (n-1)\hat{\theta}_i, \quad i = 1, \ldots, n \tag{2.20}$$

so the bias-corrected estimate of θ is the mean of the pseudo-values, $\hat{\theta}_J = \bar{u} = n^{-1} \sum_i u_i$ or equivalently,

$$\hat{\theta}_J = n\hat{\theta}(Y, m) - (n-1)\bar{\theta}, \tag{2.21}$$

where

$$\bar{\theta} = \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_i. \tag{2.22}$$

The jackknife standard error estimate for $\hat{\theta}_J$ is

$$\hat{\sigma}_{\theta_J} = \left(\left(\sum_{i=1}^{n} (u_i - \bar{u})^2 \right) \Big/ (n(n-1)) \right)^{1/2}. \tag{2.23}$$

The bootstrap provides another computational method for estimating the standard errors. In the case of censored samples, we would need to draw random samples conditional on m, the number of fully observed values and this is more awkward. The Jackknife method seems more expedient and direct in this case and if properly implemented there is unlikely to be any noticeable difference between the two methods for the purpose of estimating the standard errors of the estimates.

Illustrative Simulations

The original jackknife algorithm had two purposes:

1. bias-corrected estimates
2. estimates of the standard errors of the parameter estimates.

Illustrative simulations were done to illustrate both of these aspects.

Bias-corrected estimates

We now investigate for the Gaussian case the bias corrected estimates for $\hat{\mu}$ and $\hat{\sigma}$ produced by the Jackknife and compare these estimates with the original MLE. From the theory of maximum likelihood estimation we know that asymptotically the MLE are consistent and first order efficient and so this implies that asymptotically the root-mean-square error (RMSE) for the MLE should perform well overall.

We focus on a sample size of $n = 100$ since this is reasonably large provided the censoring is not too extreme. Samples were drawn from a normal distribution with mean 0 and variance 1. The detection point c varied between -2 and 0 corresponding to a censoring rate from about 2.2% to 50%. For each random sample we found the MLE using the EM algorithm and then applied the Jackknife.

$N = 10^5$ simulations were done. The bias in the MLE and Jackknife estimators for μ and σ is compared in Figure 2.3. The maximum standard deviation in the bias estimates was about 0.001. The bias was quite small for the MLE and negligible for the Jackknife estimate. As the censoring level increases the bias in the MLE for μ increases while for σ it slightly decreases. As might be expected when the censoring effect is negligible with $c = -2$, the bias in the MLE is also negligible for $\hat{\mu}$ but not for $\hat{\sigma}$. As expected using the jackknife reduces the bias but as shown in Table 2.1 the Jackknifed estimates are not better in terms of overall RMSE accuracy. This was not unexpected since it Jackknifed estimates are do not seem to be widely used in practice.

Estimates of the standard errors of the parameter estimates

The accuracy of the Jackknife method for estimating the standard errors of the MLE $\hat{\mu}$ and $\hat{\sigma}$ was also examined. For this purpose we estimated the empirical variances by simulation,

$$s_\mu^2 = N^{-1} \sum_{i=1}^{N} \left(\hat{\mu}_{\text{MLE}}^{(i)} - \mu \right)^2, \tag{2.24}$$

and

$$s_\sigma^2 = N^{-1} \sum_{i=1}^{N} \left(\hat{\sigma}_{\text{MLE}}^{(i)} - \sigma \right)^2, \tag{2.25}$$

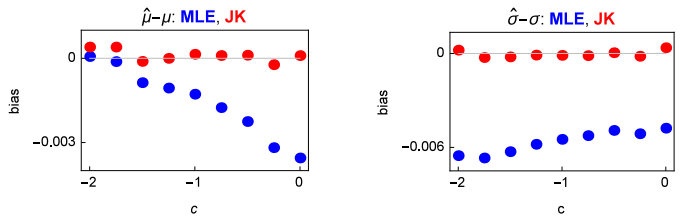

Figure 2.3: The bias of the MLE and jackknife estimators for the mean and standard deviation are compared.

c	rmse($\hat{\mu}_{\text{MLE}}$)	rmse($\hat{\mu}_{\text{JK}}$)	rmse($\hat{\sigma}_{\text{MLE}}$)	rmse($\hat{\sigma}_{\text{JK}}$)
-2.00	0.100	0.100	0.072	0.072
-1.75	0.100	0.100	0.074	0.074
-1.50	0.101	0.100	0.075	0.075
-1.25	0.101	0.101	0.077	0.077
-1.00	0.102	0.102	0.080	0.081
-0.75	0.104	0.104	0.086	0.086
-0.50	0.108	0.107	0.092	0.092
-0.25	0.114	0.114	0.100	0.100
0.00	0.125	0.124	0.112	0.113

Table 2.1: RMSE comparisons of Jackknife estimates with MLE.

where $\hat{\mu}^{(i)}_{\text{MLE}}$ and $\hat{\sigma}^{(i)}_{\text{MLE}}$ for $i = 1, \ldots, N$ denote the MLE estimates in the ith simulation. These empirical estimates were compared to the average of the Jackknife estimators,

$$\bar{\sigma}_{\mu} = N^{-1} \sum_{i=1}^{N} \hat{\sigma}^{(i)}_{\mu}, \tag{2.26}$$

and

$$\bar{\sigma}_{\sigma} = N^{-1} \sum_{i=1}^{N} \hat{\sigma}^{(i)}_{\sigma}. \tag{2.27}$$

As shown in Table 2.2. the agreement is quite close confirming the usefulness of the jackknife method for estimating the standard errors of the parameter estimates.

Multiple-censored samples

We assume a random sample $y_1, \ldots, y_n$ has been observed with censoring process $c_i = \left(c_i^-, c_i^+\right), i = 1, \ldots, n$. We assume that the underlying latent process $Z_1, \ldots, Z_n$ is normally distributed with parameters μ and σ, that denote the mean and standard deviation. As remarked in Chapter 1, if the ith observation is left-censored, the underlying latent variable conditional on this, will have a right-truncated distribution and similarly with right-censoring, the corresponding conditional distribution is left-truncated. Let $\mathrm{E}_Z\left\{Z \left| \hat{\mu}^{(j)}, \hat{\sigma}^{(j)}, c_i^- \right.\right\}$ and $\mathrm{E}_Z\left\{Z \left| \hat{\mu}^{(j)}, \hat{\sigma}^{(j)}, c_i^+ \right.\right\}$ denote the expectations in the corresponding right and left truncated distributions.

c	s_μ	$\bar{\sigma}_\mu$	s_σ	$\bar{\sigma}_\sigma$
-2.00	0.100	0.100	0.072	0.072
-1.75	0.100	0.100	0.073	0.073
-1.50	0.101	0.100	0.075	0.074
-1.25	0.101	0.101	0.077	0.077
-1.00	0.102	0.102	0.080	0.080
-0.75	0.104	0.104	0.085	0.085
-0.50	0.108	0.107	0.091	0.091
-0.25	0.114	0.114	0.100	0.100
0.00	0.125	0.125	0.112	0.111

Table 2.2: Comparing Jackknife estimates for the standard errors with empirical simulation estimates.

The following algorithm covers the general case but Algorithm CM1 is faster and simpler in the case of singly-censored data.

Algorithm CM. General algorithm for censored MLE estimation.

Step 1: Initialization:

$$u_t \longleftarrow \begin{cases} y_t & \text{if } y_t\text{not censored;} \\ \mathrm{E}_Z\left\{Z \left| \hat{\mu}^{(j)}, \hat{\sigma}^{(j)}, c_i^+ \right.\right\} & \text{left truncated case;} \\ \mathrm{E}_Z\left\{Z \left| \hat{\mu}^{(j)}, \hat{\sigma}^{(j)}, c_i^- \right.\right\} & \text{right truncated case;} \end{cases} \tag{2.28}$$

Step 2: Set $\hat{\mu}^{(j+1)} \longleftarrow \bar{u}$

Step 3:

$$v_t \longleftarrow \begin{cases} (y_t - \hat{\mu})^2 & \text{if } y_t\text{not censored;} \\ \mathrm{E}_Z\left\{\left(Z - \hat{\mu}^{(j)}\right)^2 | \hat{\mu}^{(j)}, \hat{\sigma}^{(j)}, c_i^+\right\} & \text{left truncated case;} \\ \mathrm{E}_Z\left\{\left(Z - \hat{\mu}^{(j)}\right)^2 | \hat{\mu}^{(j)}, \hat{\sigma}^{(j)}, c_i^-\right\} & \text{right truncated case;} \end{cases} \tag{2.29}$$

Step 4: $\left(\hat{\sigma}^2\right)^{(j+1)} \longleftarrow \bar{v}$

Step 5: Test for convergence of the estimates. If they have not converged, $j \longleftarrow j + 1$ and repeat Steps 1 to 4.

This algorithm is used in our algorithm CENLTSA for fitting linear time series models to censored time series.

Simulation experiment to compare single and double censoring

A brief simulation experiment was done to illustrate the performance of the CM algorithm and to compare singly-censored samples with doubly-censored samples in very different censoring regimes. The latent process, $Z_t, t = 1, \ldots, 60$ where $Z_t \sim \mathrm{NID}(100, 15)$ was used. In the single

left-censoring case $c = 115$ was used which corresponds to a very high censoring rate. In the double left-censoring case, $c_1 = 115$, was used for half the data but $c_2 = 70$ was used for the other half. So there is much more information provided in the doubly censored regime that we used. This is reflected in the improved accuracy of the estimates when the boxplots of the estimates based on 10^3 simulations are compared shown in Figure 2.4. The doubly censored regime provided more information resulting in improved accuracy of the estimates.

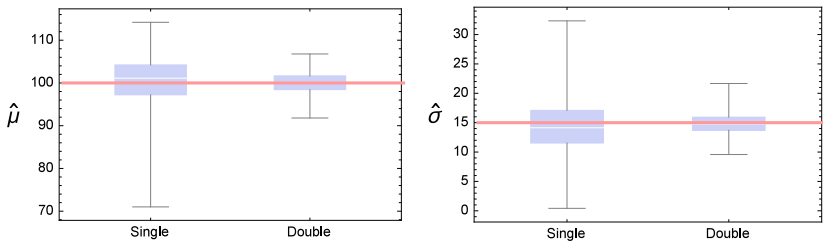

Figure 2.4: Boxplots of the estimates in singly and doubly censored sampling simulated example.

Information matrix

Information matrix for censored normal samples

In general, under regularity conditions, the MLE estimates, suitably normalized, weakly converge to a normal distribution with covariance matrix equal to the inverse of the information matrix. Thus the information matrix is useful in estimating the approximate standard errors of the MLE estimates.

For comparison with the censored case, we first give the result for random sampling from a complete normal distribution. As discussed in Knight [2000, Example 5.14, p.258], it is simpler to work with the information matrix for (μ, σ) rather than $\left(\mu, \sigma^2\right)$. In the normal IID case with complete data for a random sample of size n from a normal population with mean μ and variance σ the Fisher information matrix for μ and σ may be written,

$$\mathcal{I}(\mu, \sigma) = \begin{pmatrix} n\sigma^{-2} & 0 \\ 0 & 2n\sigma^{-2} \end{pmatrix}. \tag{2.30}$$

We now obtain the observed and expected information matrix in the more general case with left-censoring using the method of Oakes [1999]. From eqn. (1.35) we may write,

$$\mathcal{I}_c(\theta, y) = -\left\{ \frac{\partial^2 Q(\theta'|\theta)}{\partial \theta'^2} + \frac{\partial^2 Q(\theta'|\theta)}{\partial \theta' \partial \theta} \right\}_{\theta'=\theta}. \tag{2.31}$$

where $\theta = (\mu, \sigma)'$ and $y = (y_1, \ldots, y_m)'$ is the vector of complete data and censor point, c. Hence let,

$$\mathcal{I}_c(\mu, \sigma, y) = \begin{pmatrix} i_{1,1} & i_{1,2} \\ i_{1,2} & i_{2,2} \end{pmatrix}, \tag{2.32}$$

$$i_{1,1} = -\left(\frac{\partial^2 Q(\mu', \sigma'|\mu, \sigma)}{\partial \mu'^2} + \frac{\partial^2 Q(\mu', \sigma'|\mu, \sigma)}{\partial \mu' \partial \mu}\right)_{(\mu',\sigma')=(\mu,\sigma)} \tag{2.33}$$

$$i_{1,2} = -\left(\frac{\partial^2 Q(\mu', \sigma'|\mu, \sigma)}{\partial \mu' \partial \sigma'} + \frac{\partial^2 Q(\mu', \sigma'|\mu, \sigma)}{\partial \mu' \partial \sigma}\right)_{(\mu',\sigma')=(\mu,\sigma)} \tag{2.34}$$

$$i_{2,2} = -\left(\frac{\partial^2 Q(\mu', \sigma'|\mu, \sigma)}{\partial (\sigma')^2} + \frac{\partial^2 Q(\mu', \sigma'|\mu, \sigma)}{\partial \sigma' \partial \sigma}\right)_{(\mu',\sigma')=(\mu,\sigma)} \tag{2.35}$$

The expected information is

$$\mathcal{I}_c(\mu, \sigma) = \begin{pmatrix} \mathrm{E}\{i_{1,1}\} & \mathrm{E}\{i_{1,2}\} \\ \mathrm{E}\{i_{1,2}\} & \mathrm{E}\{i_{2,2}\} \end{pmatrix} \tag{2.36}$$

Derivation of $i_{1,1}$

From eqn. (2.12) we can write,

$$\frac{\partial Q}{\partial \mu'} = \sigma^{-2\prime} \left\{ \sum_{i=1}^{m} (y_i - \mu') + (n-m)\mathrm{E}_{Z(\mu,\sigma,c)}\{(Z - \mu')\} \right\}. \tag{2.37}$$

So

$$\frac{\partial Q}{\partial \mu'^2} = -n\sigma^{-2} \tag{2.38}$$

and

$$\frac{\partial^2 Q}{\partial \mu' \partial \mu} = (n-m)\sigma^{-2}\partial_\mu \mathrm{E}_{Z(\mu,\sigma,c)}\{Z\}. \tag{2.39}$$

Using *Mathematica*,

$$\partial_\mu \mathrm{E}_{\{\mu,\sigma\}}\{Z\} = \frac{1}{\pi\sigma\, \mathrm{erfc}\left(\frac{\mu-c}{\sqrt{2}\sigma}\right)^2} e^{-\frac{(c-\mu)^2}{\sigma^2}} (\mathcal{A}_1 + \mathcal{A}_2), \tag{2.40}$$

where

$$\mathcal{A}_1 = \sqrt{2\pi} e^{\frac{(c-\mu)^2}{2\sigma^2}} \left(\mu\, \mathrm{erf}\left(\frac{c-\mu}{\sqrt{2}\sigma}\right) - c\, \mathrm{erfc}\left(\frac{\mu-c}{\sqrt{2}\sigma}\right) + \mu\right)$$

and

$$\mathcal{A}_2 = \pi\sigma e^{\frac{(c-\mu)^2}{\sigma^2}} \left(\mathrm{erf}\left(\frac{c-\mu}{\sqrt{2}\sigma}\right) + 1\right)^2 - 2\sigma.$$

Hence eqns. (2.38) and (2.39) we can write,

$$i_{1,1} = n\sigma^{-2} - (n-m)\sigma^{-2}\partial_\mu \mathrm{E}_{Z(\mu,\sigma,c)}\{Z\}. \tag{2.41}$$

where $\partial_\mu \mathrm{E}_z\{Z\}$ may be computed using eqn. (2.40). Notice that the information on μ is comprised of two components. One component, $\sigma^{-2}n$, corresponds to the case of n complete observations and then this is adjusted by subtracting $\sigma^{-2}(n-m)\partial_\mu \mathrm{E}_z\{Z\}$ is due to the censored

observations. The expected information, $\mathrm{E}\{i_{1,1}\}$, is obtained by replacing m by its expected value, $n_m = n(1 - \Phi(c; \mu, \sigma))$, in eqn. (2.41).

Figure 2.5 shows how the expected information on μ depends on c in the case of the standard normal distribution. Due to left-censoring, it is not symmetric about $c = 0$.

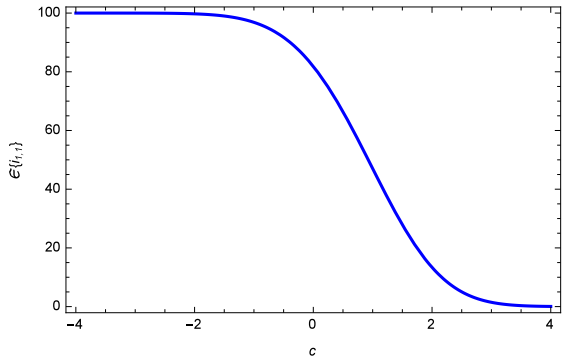

Figure 2.5: The expected information for the mean in left-censored samples of size 100.

Derivation of $i_{1,2}$

Next we obtain the off-diagonal elements for $i_{1,2}$ in eqn. (2.34). Recall from (2.37) we have,

$$\frac{\partial Q}{\partial \mu'} = \sigma^{-2\prime} \left\{ \sum_{i=1}^{m} (y_i - \mu') + (n-m)\mathrm{E}_{Z(\mu,\sigma,c)} \{(Z - \mu')\} \right\}. \tag{2.42}$$

Hence differentiating with respect to σ',

$$\frac{\partial Q}{\partial \mu' \partial \sigma'} = -2\sigma^{-3\prime} \left(\sum_{i=1}^{m} (y_i - \mu') + (n-m) \left(\mathrm{E}_{Z(\mu,\sigma,c)}\{Z\} - \mu' \right) \right). \tag{2.43}$$

Next, differentiating eqn. (2.37) with respect to σ,

$$\frac{\partial^2 Q}{\partial \mu' \partial \sigma} = \sigma^{-2\prime}(n-m)\, \partial_\sigma \mathrm{E}_{Z(\mu,\sigma,c)}\{Z\}, \tag{2.44}$$

where $\partial_\sigma \mathrm{E}_Z\{Z\}$ may be determined using *Mathematica*,

$$\partial_\sigma \mathrm{E}_z\{Z\} = e^{-\frac{(c-\mu)^2}{\sigma^2}} \left(\sqrt{2\pi} e^{\frac{(c-\mu)^2}{2\sigma^2}} \left((c-\mu)^2 + \sigma^2 \right) \left(\mathrm{erfc}\left(\frac{c-\mu}{\sqrt{2}\sigma} \right) - 2 \right) + 2\sigma(\mu - c) \right) / \left(\pi\sigma^2 \mathrm{erfc}\left(\frac{\mu - c}{\sqrt{2}\sigma} \right)^2 \right). \tag{2.45}$$

From eqns. (2.43) and (2.44),

$$i_{1,2} = 2\sigma^{-3} \left(\sum_{i=1}^{m} (y_i - \mu) + (n-m) \left(\mathrm{E}_z\{Z\} - \mu \right) \right) - \sigma^{-2}(n-m)\, \partial_\sigma \mathrm{E}_{Z(\mu,\sigma,c)}\{Z\} \tag{2.46}$$

It may be seen that $\mathrm{E}\{i_{1,2}\} \to 0$ as $c \to -\infty$. This follows from the fact the $m \to n$ as $c \to -\infty$ and $\mathrm{E}\{\mu - \mu\} = 0$.

To obtain the expected information first we consider the term,

$$S = \sum_{i=1}^{M} (Y_i - \mu) \tag{2.47}$$

where Y_i are left-truncated random variables from the normal distribution (μ, σ) truncated on (c, ∞) and M is binomially distributed with parameters n and $p = 1 - \Phi(c)$. Then by the law of total expectation we have, $\mathrm{E}\{S\} = \mathrm{E}_M\{\mathrm{E}_{S|M=m}\{S\}\} = \mathrm{E}_M(M)(\mathrm{E}_Y\{Y\} - \mu)$, where Y is the left-truncated normal distribution with parameters (μ, σ, c). Then $\mathrm{E}_M(M) = n(1 - \Phi(c))$ and $\mu_y = \mathrm{E}_Y\{Y\}$ may be obtained using *Mathematica*,

$$\mu_y = \mu + \sqrt{\frac{2}{\pi}}\sigma e^{-\frac{(\mu+1)^2}{2\sigma^2}} / \left(1 + \mathrm{erf}\left(\frac{\mu+1}{\sqrt{2}\sigma}\right)\right) \tag{2.48}$$

Then m is replaced by its expected value, $n_m = n(1 - \Phi(c; \mu, \sigma))$. Hence,

$$\mathrm{E}\{i_{1,2}\} = 2\sigma^{-3}\left(n_m\left(\mu_y - \mu\right) + (n - n_m)\left(\mathrm{E}_{\{\mu,\sigma\}}\{Z\} - \mu\right)\right) - \sigma^{-2}(n - n_m)\ \partial_\sigma \mathrm{E}_{Z(\mu,\sigma,c)}\{Z\} \tag{2.49}$$

Figure 2.6 illustrates how $\mathrm{E}\{i_{1,2}\}$ depends on c in the standard normal case. As expected as c decreases, $i_{1,2} \to 0$. As c increases, $\mathrm{E}\{i_{1,2}\}$ reaches a maximum value of about 56.1 when $c \approx 0.84$ and afterwards it decreases to zero. When $c < -2$, the parameters are nearly orthogonal as is the case when $c > 4$ but for $-2 < c < 4$ when may expect that the estimates will be correlated.

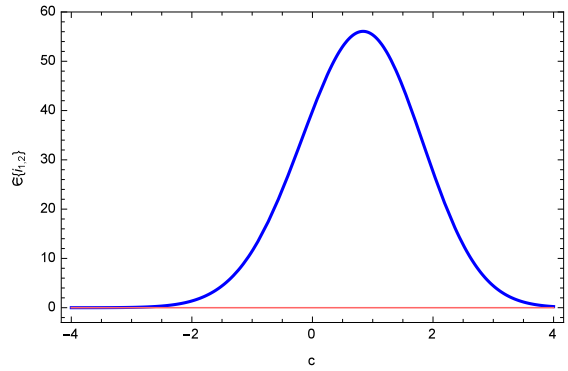

Figure 2.6: The expected joint information the mean and standard deviation in left-censored N(0,1) samples of size 100.

Derivation of $i_{2,2}$

To obtain next term in the information matrix, $i_{2,2}$, we need to evaluate $\partial Q \big/ \partial (\sigma')^2$ and $\partial Q / \partial \sigma'\ \partial \sigma$.

$$Q(\mu', \sigma' | \mu, \sigma, y) = -n \log(\sigma') - \frac{1}{2\sigma'^2}\left\{\sum_{i=1}^{m} (y_i - \mu')^2 + (n - m)\mathrm{E}_Z\left\{(Z - \mu')^2\right\}\right\} \tag{2.50}$$

From above,

$$\frac{\partial Q}{\partial \sigma'} = -\frac{n}{\sigma'} + (\sigma')^{-3}\left(\sum_{i=1}^{m}(y_i - \mu')^2 + (n-m)\mathrm{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu')^2\right\}\right) \tag{2.51}$$

Taking the second partial derivative with respect to σ',

$$\frac{\partial^2 Q}{\partial \sigma' \partial \sigma'} = \frac{n}{(\sigma')^2} - 3(\sigma')^{-4}\left(\sum_{i=1}^{m}(y_i - \mu')^2 + (n-m)\mathrm{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu')^2\right\}\right), \tag{2.52}$$

Next, differentiating eqn. (2.51) with respect to σ,

$$\frac{\partial^2 Q}{\partial \sigma' \partial \sigma} = (n-m)(\sigma')^{-3}\,\partial_\sigma \mathrm{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu')^2\right\}. \tag{2.53}$$

From eqns. (2.52) and (2.53) and evaluating at $(\mu', \sigma') = (\mu, \sigma)$,

$$i_{2,2} = -n\Big/\sigma^2 + 3\sigma^{-4}\left(\sum_{i=1}^{m}(y_i - \mu)^2 + (n-m)\mathrm{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu')^2\right\}\right) - (n-m)\sigma^{-3}\partial_\sigma \mathrm{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu)^2\right\}. \tag{2.54}$$

Explicit formulas for $\mathrm{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu')^2\right\}$ and $\partial_\sigma \mathrm{E}_{Z(\mu,\sigma,c)}\{(Z-\mu')\}^2$ may be obtained using *Mathematica* symbolics,

$$\mathrm{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu')^2\right\} = \frac{1}{\sigma^2 \mathrm{Erfc}\left[\frac{-c+\mu}{\sqrt{2}\sigma}\right]^2}(\mathcal{B}_1 + \mathcal{B}_2), \tag{2.55}$$

where

$$\mathcal{B}_1 = \left(e^{\frac{(c-\mu)^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}\left(1 + \mathrm{Erf}\left[\frac{c-\mu}{\sqrt{2}\sigma}\right]\right)\left(\sigma^2 + (\mu-\mu')^2\right) - \frac{2\sigma(c+\mu-2\mu')}{\pi}\right)$$

and

$$\mathcal{B}_2 = \mathrm{Erfc}\left[\frac{-c+\mu}{\sqrt{2}\sigma}\right] \left(2\sigma^3\left(1 + \mathrm{Erf}\left[\frac{c-\mu}{\sqrt{2}\sigma}\right]\right) - e^{-\frac{(c-\mu)^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}(c-\mu')\left(c^2 + 2\sigma^2 + \mu\mu' - c(\mu+\mu')\right)\right)$$

$$\partial_\sigma \mathrm{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu')^2\right\} = \frac{1}{\sigma^2 \mathrm{Erfc}\left[\frac{-c+\mu}{\sqrt{2}\sigma}\right]^2}(C_1 + C_2), \tag{2.56}$$

where

$$C_1 = e^{-\frac{(c-\mu)^2}{\sigma^2}}(c-\mu)\left(e^{\frac{(c-\mu)^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}\left(1 + \mathrm{Erf}\left[\frac{c-\mu}{\sqrt{2}\sigma}\right]\right)\left(\sigma^2 + (\mu-\mu')^2\right) - \frac{2\sigma(c+\mu-2\mu')}{\pi}\right)$$

and

$$C_2 = \text{Erfc}\left[\frac{-c+\mu}{\sqrt{2}\sigma}\right] \left(2\sigma^3\left(1+\text{Erf}\left[\frac{c-\mu}{\sqrt{2}\sigma}\right]\right) - e^{-\frac{(c-\mu)^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}(c-\mu')\left(c^2+2\sigma^2+\mu\mu'-c(\mu+\mu')\right)\right).$$

To obtain the expected information, $\text{E}\{i_{2,2}\}$ first we consider the term,

$$T = \sum_{i=1}^{M}(Y_i-\mu)^2 \tag{2.57}$$

where Y_i are left-truncated random variables from the normal distribution (μ,σ) truncated on (c,∞) and M is binomially distributed with parameters n and $p = 1-\Phi(c)$. Then by the law of total expectation we have, $\text{E}\{T\} = \text{E}_M\{\text{E}_{S|M=m}\{T\}\} = \text{E}_M(M)\,\text{E}_{Y(\mu,\sigma,c)}\left\{(Y-\mu)^2\right\}$, where $\text{E}_M(M) = n(1-\Phi(c;\mu,\sigma)$ and $\Phi(c;\mu,\sigma)$ denotes the cumulative distribution of the normal distribution with mean μ and standard deviation σ evaluated at c and Y is the left-truncated normal distribution with parameters (μ,σ,c). So the probability density function for Y is proportional to the normal probability density function with mean μ and standard deviation on the interval (c,∞) and zero outside this interval. The constant of proportionality is determined so that the truncated probability density function integrates to one.

Using *Mathematica*,

$$\text{E}_{Y(\mu,\sigma,c)}\left\{(Y-\mu)^2\right\} = \frac{\text{Erfc}\left[\frac{c-\mu}{\sqrt{2}\sigma}\right]\left(\sigma^2+(\mu-\mu')^2\right) + e^{-\frac{(c-\mu)^2}{2\sigma^2}}\sqrt{\frac{2}{\pi}}\sigma(c+\mu-2\mu')}{1+\text{Erf}\left[\frac{-c+\mu}{\sqrt{2}\sigma}\right]} \tag{2.58}$$

Hence,

$$\text{E}\{i_{2,2}\} = -n\Big/\sigma^2 + 3\sigma^{-4}\left(n_m\text{E}_{Y(\mu,\sigma,c)}\left\{(Y-\mu)^2\right\} + (n-n_m)\,\text{Var}(Z)\right) - (n-n_m)\,\sigma^{-3}\partial_\sigma\text{E}_{Z(\mu,\sigma,c)}\left\{(Z-\mu)^2\right\}. \tag{2.59}$$

As a check on eqn. (2.59), $N = 10^4$ random samples of size 100 from the left-truncated normal distribution with parameters $\mu = 0$ and $\sigma = 1$ were used to estimate the average value of $i_{2,2}$ using eqn. (2.59) for eleven truncation points $-2.5, -2, \ldots, 2, 2.5$.

Figure 2.7 plots $\text{E}\{i_{2,2}\}$ and the eleven empirical means. We see the agreement is good confirming the correctness of eqn. (2.59).

As c increases there is less information as may be expected. The shape of the curve is a little surprising since we would have anticipated a more steady rate of decrease but the curve is almost flat when $c \in (0.5, 1)$.

Test data example

As a check on the algorithm for the information matrix an example test data set was used to compare the results given by our algorithm with those produced using the algorithms of

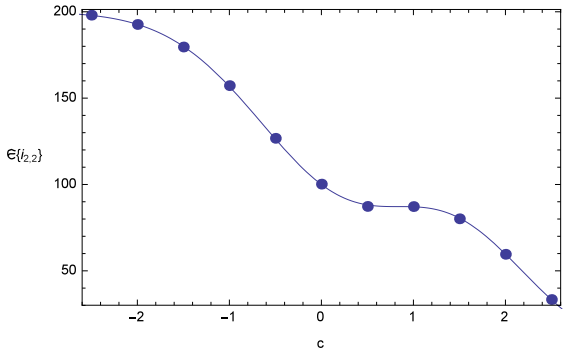

Figure 2.7: The expected information for the standard deviation in left-censored samples.

Wolynetz [1979a], Henningsen [2012], Henningsen and Toomet [2011]. Wolynetz [1979a] derives the observed Fisher information matrix using the log-likelihood function of the data, $L(\mu, \sigma|\text{data}, c)$ but the equivalence of the formula derived with our result is not easy to see algebraically. Henningsen [2012] uses the Hessian of the log-likelihood function to obtain an approximation to the observed information. For our test data we took,

$$y = (-2, -2, -2, -1, -1, -1, 0, 0, 0, 1, 1, 1, 2, 2, 2) \tag{2.60}$$

with $c = -1.5$. Our algorithm produced the following estimate for the covariance matrix of $(\hat{\mu}, \hat{\sigma})$. This result agrees the algorithms of Wolynetz (1979) and Henningsen (2012).

$$\begin{pmatrix} 0.16834362 & -0.01684593 \\ -0.01684593 & 0.11021454 \end{pmatrix} \tag{2.61}$$

The maximum likelihood estimate for the parameters was $\hat{\mu} = -0.06662881$ and $\hat{\sigma} = 1.54378019$. Our R package cents McLeod and Mohammad [2012] contains implements the Fortran algorithm given in Wolynetz (1979) gives an example using this test dataset.

Large sample properties of maximum likelihood estimator

The large-sample covariance matrix for the maximum likelihood estimates is

$$\mathcal{I}(\mu, \sigma)^{-1} = n^{-1} \begin{pmatrix} \sigma_{\mu,\mu} & \sigma_{\mu,\sigma} \\ \sigma_{\mu,\sigma} & \sigma_{\sigma,\sigma} \end{pmatrix} \tag{2.62}$$

where $\sigma_{\mu,\mu}$ and $\sigma_{\sigma,\sigma}$ are the large-sample variances of $\hat{\mu}$ and $\hat{\sigma}$ per observation and n is the number of observations.

Taking $n = 1$ and $\sigma = 1$ in eqn (2.62), the large sample variance of the maximum likelihood estimators for μ and σ^2 in the case of complete normal samples is 1 and 0.5 respectively. In left-censored samples this corresponds to the censor rate, $r = 0$. Figure 2.8 shows that for the left-censored case, $\sigma_\mu^2 = \sigma_{\mu,\mu}$ and $\sigma_\sigma^2 = \sigma_{\sigma,\sigma}$, obtained from the inverse of the expected information matrix as a function of r, the censor rate. When $r > 0.7$ both these variances rapidly increase due to the fact the there is less and less information about the true mean and variance in the censored sample.

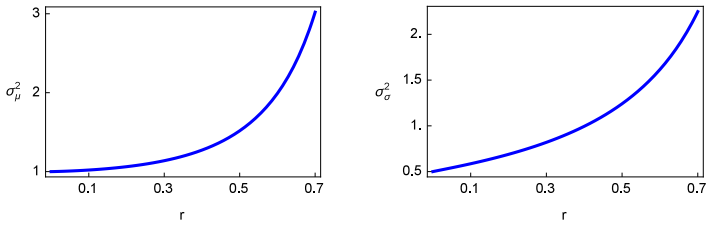

Figure 2.8: Asymptotic variances of censored MLE for mean and standard deviation.

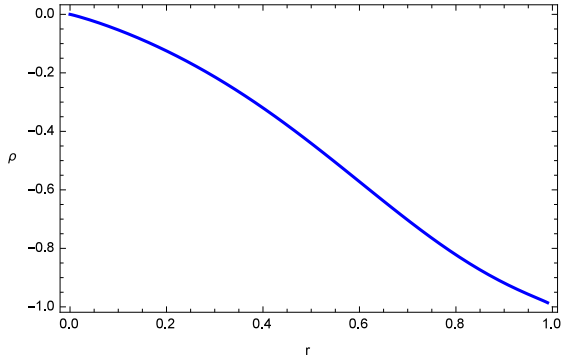

Figure 2.9: Asymptotic correlation between MLE estimate for mean and standard deviation in left-censored normal samples.

Figure 2.9 shows the correlation $\rho = \sigma_{\mu,\sigma}/\left(\sigma_\mu \sigma_\sigma\right)$ as a function of r. As r increases the maximum likelihood estimates $\hat{\mu}$ and $\hat{\sigma}$ become more and more negatively correlated.

In large censored samples the maximum likelihood estimators, $\hat{\mu}$ and $\hat{\sigma}$, have sampling distribution that is bivariate normal with mean (μ, σ) and covariance matrix $n^{-1}\mathcal{I}(\mu, \sigma)^{-1}$, where n is the sample size. The ellipsoids of concentration corresponding 95% and 50% probability for censor rates, $r = 0.1, 0.3, 0.5$ and 0.7, are shown in Figure 2.10. As r increases the ellipse becomes more elongated the area of region increases.

Comparing expected and empirical standard deviations

The large-sample variance matrix for the parameters is estimated by

$$\mathcal{I}(\mu, \sigma)^{-1} = n^{-1} \begin{pmatrix} \sigma_{\mu,\mu} & \sigma_{\mu,\sigma} \\ \sigma_{\mu,\sigma} & \sigma_{\sigma,\sigma} \end{pmatrix} \tag{2.63}$$

where $\sigma_{\mu,\mu}$ and $\sigma_{\sigma,\sigma}$ are the large-sample variances of $\hat{\mu}$ and $\hat{\sigma}$ per observation and n is the number of observations. The empirical variances of the estimates $\hat{\mu}$ and $\hat{\sigma}$ may be obtained by simulation.

In the simulation, a random sample of size $n = 100$ was generated as normally and independently distributed with mean zero and variance one and then the sample was left-censored with censor point $c_j = \Phi^{-1}\left(r_j\right)$, where Φ^{-1} denotes the inverse cumulative distribution function for the standard normal and r_j, $j = 1, \ldots, 7$ is the censor rate where r_j is the jth element of the vector $r = (0.01, 0.2, \ldots, 0.7)$. For each $j = 1, \ldots, 7$, $N = 10^4$ simulations were done and the

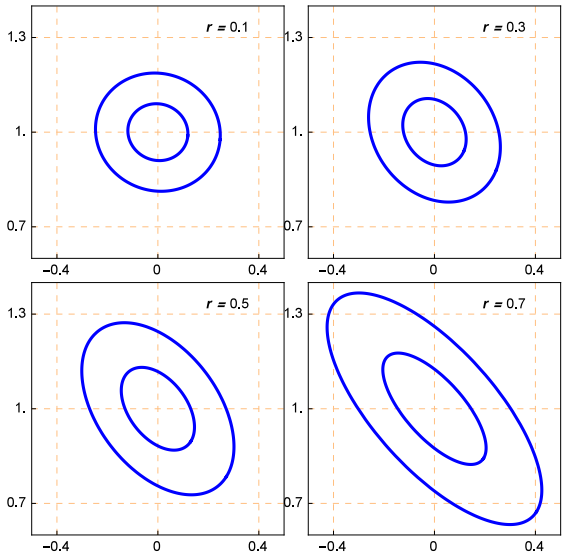

Figure 2.10: Ellipsoids of concentration corresponding to 0.95 and 0.5 probability for four censor rates. In each panel, the vertical axes corresponds to the standard deviation and the horizontal axis to the mean.

empirical standard errors for the maximum likelihood estimates for μ and σ were obtained,

$$\hat{\sigma}_{\mu}^{(j)} = \sqrt{N^{-1} \sum_{i=1}^{N} \hat{\mu}_{i,j}^2} \tag{2.64}$$

$$\hat{\sigma}_{\sigma}^{(j)} = \sqrt{N^{-1} \sum_{i=1}^{N} \left(\hat{\sigma}_{i,j} - 1\right)^2} \tag{2.65}$$

where $\hat{\mu}_{i,j}$ and $\hat{\sigma}_{i,j}$ are the maximum likelihood estimates in the ith simulation with parameter setting r_j, $j = 1, \ldots, 7$; $i = 1, \ldots, N$. The standard deviations for $\hat{\sigma}_{\mu}^{(j)}$ and $\hat{\sigma}_{\sigma}^{(j)}$ are approximately

$$\text{est.sd}\left(\hat{\sigma}_{\mu}^{(j)}\right) \approx \hat{\sigma}_{\mu}^{(j)} / \sqrt{2N} \tag{2.66}$$

and

$$\text{est.sd}\left(\hat{\sigma}_{\sigma}^{(j)}\right) \approx \hat{\sigma}_{\sigma}^{(j)} / \sqrt{2N} \tag{2.67}$$

which were quite small due to choosing N fairly large. It is convenient to normalize these values on a per observation basis by multiplying them by $\sqrt{n}$. Let $\tilde{\sigma}_{\mu}^{(j)} = \sqrt{n}\hat{\sigma}_{\mu}^{(j)}$ and $\tilde{\sigma}_{\sigma}^{(j)} = \sqrt{n}\hat{\sigma}_{\sigma}^{(j)}$.

The theoretical expected values corresponding to $\tilde{\sigma}_{\mu}^{(j)}$ and $\tilde{\sigma}_{\sigma}^{(j)}$ from eqn. (2.63) are $\sigma_{\mu} = \sqrt{\sigma_{\mu,\mu}}$ and $\sigma_{\sigma} = \sqrt{\sigma_{\sigma,\sigma}}$ and these compared in the Tables 2.3 and 2.4. Figure 2.11 shows the theoretical expected values as the solid curve and points correspond to the empirical estimates.

We conclude that the expected information matrix provides an accurate approximation over the range of censor rates from 0 to 70% and for sample sizes $n \geq 100$.

c	$\hat{\sigma}_\mu$(th)	$\hat{\sigma}_\mu$(em)	sd($\hat{\sigma}_\mu$(em))
0.1	1.010	1.016	0.007
0.2	1.031	1.020	0.007
0.3	1.067	1.075	0.008
0.4	1.128	1.141	0.008
0.5	1.232	1.267	0.009
0.6	1.411	1.416	0.010
0.7	1.738	1.804	0.013

Table 2.3: Comparing the asymptotic approximation for the estimated standard error of the censored MLE for mean with the estimate obtained empirically by simulation. The standard deviation of the empirical estimate is shown in the last column.

c	$\hat{\sigma}_\sigma$(th)	$\hat{\sigma}_\sigma$(em)	sd($\hat{\sigma}_\sigma$(em))
0.1	0.765	0.753	0.005
0.2	0.830	0.825	0.006
0.3	0.905	0.900	0.006
0.4	0.997	1.007	0.007
0.5	1.114	1.133	0.008
0.6	1.271	1.282	0.009
0.7	1.499	1.527	0.011

Table 2.4: Comparing the asymptotic approximation for the estimated standard error of the censored MLE for the standard deviation with the estimate obtained empirically by simulation. The standard deviation of the empirical estimate is shown in the last column.

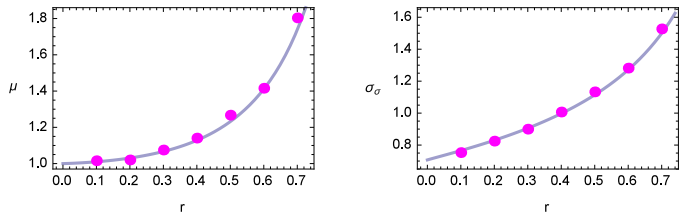

Figure 2.11: Comparing asymptotic standard error for MLE for mean and standard deviation with empirical estimates based on simulation.

Robust dynamic graphical estimation and diagnostic checking

We propose a new dynamic graphical method that provides robust estimates. This method is explained in detail for the censored normal case.

The dynamic normal probability plot provides an interactive graphical method for robust estimation of the mean and standard deviation parameters, μ and σ, from a random sample of size n with possible censoring. We suppose that we have m fully observed values $Y_1, \ldots, Y_m$ with corresponding censor points $c_1, \ldots, c_m$ and that the underlying latent variables are independent and identically distribution normal random variables, $Z_1, \ldots, Z_n$. Then we set $Y_i = \max(c_i, Z_i), i = 1, \ldots, n$ in the case of left-censoring where without loss of generality we assume that $Z_i > c_i, i = 1, \ldots, m$ and $Z_i \leq c_i, i = m+1, \ldots, n$. In the right-censoring case, $Y_i = \min(c_i, Z_i)$ and the inequalities are simply reversed. In the left-censoring case $Y_1, \ldots, Y_m$ are a random sample from a left-truncated normal distribution on $(c_i, \infty), i = 1, \ldots, m.$ whereas in the right-censoring case, $Y_1, \ldots, Y_m$ are from a right-truncated distribution on $(-\infty, c_i), i = 1, \ldots, m$.

For the truncated normal we may assume $Y_1 \leq Y_2 \leq \ldots \leq Y_m$ and plot these data quantiles against the corresponding normal quantiles for the truncated distribution, say $\mathcal{Z}_1, \ldots, \mathcal{Z}_m$, where $\mathcal{Z}_i = \Phi^{-1}(p_i; \mu, \sigma, (c_i, \infty))$ in the left-censored or right-truncated case or $\mathcal{Z}_i = \Phi^{-1}(p_i; \mu, \sigma, (-\infty, c_i))$ in the right-censored or left-truncated case, where Φ^{-1} denotes the inverse normal distribution function for the truncated distribution with the indicated parameters and $p_i = (i - 0.5)/m, i = 1, \ldots, m$.

Mohammad and McLeod [2013] illustrate the use of the dynamic normal probability plot with singly-censored samples from normal or scaled t_4 distributions. Figure 2.12 illustrates how this method works to provide more robust estimates of μ and σ in the case of the t_4 distribution which has be scaled so that $\mu = 100$ and $\sigma = 15$. In this case Gaussian MLE yields, $\hat{\mu} = 101.89$ and $\hat{\sigma} = 33.85$. Using the interactive controls we fit that bulk of the data lie on a line determined by $\tilde{\mu} = 97.4$ and $\tilde{\sigma} = 14.05$. In this case the MLE estimate of the mean is slightly more accurate but the robust estimate of σ is much more accurate than the MLE.

Industrial life-testing

Data from an experiment on the life spans in thousands of miles for $n = 96$ electronic locomotive engines are given in Cohen [1991, Example 2.7.3]. These data were right-censored with $c = 135$ and $m = 37$ complete observations were obtained. A log to the base 10 transformation was used on the data. Cohen [1991] obtained parameter estimates and their standard errors, $\hat{\mu} = 2.224 \pm 0.0458$ and $\hat{\sigma} = 0.307 \pm 0.0410$. Using our MLE algorithm, we obtained $\hat{\mu} = 2.222$ and $\hat{\sigma} = 0.309$. The Jackknife estimates for the standard errors was $\sigma_{\hat{\mu}} = 0.046$ and $\sigma_{\hat{\sigma}} = 0.044$. The dynamic normal diagnostic plot is shown in Figure 2.13 does not suggest any model inadequacy.

Water quality example

The dynamic normal probability plot of the logged data reveals suggests that there has been a shift in the mean of the time series after the detection level was lowered. The ordering of first and second in the plot corresponds to data at the lowest detection value and larger detection

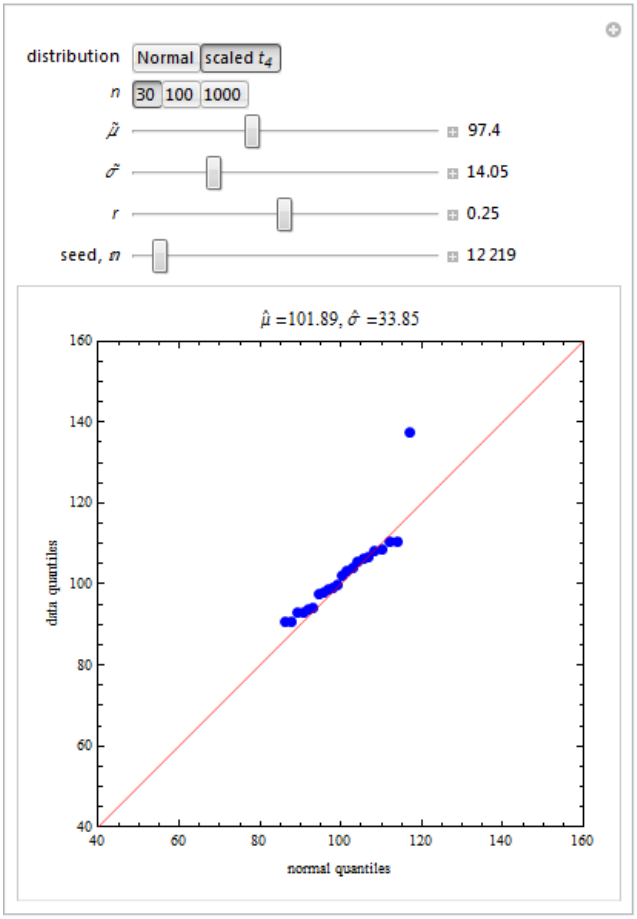

Figure 2.12: Dynamic normal plot.

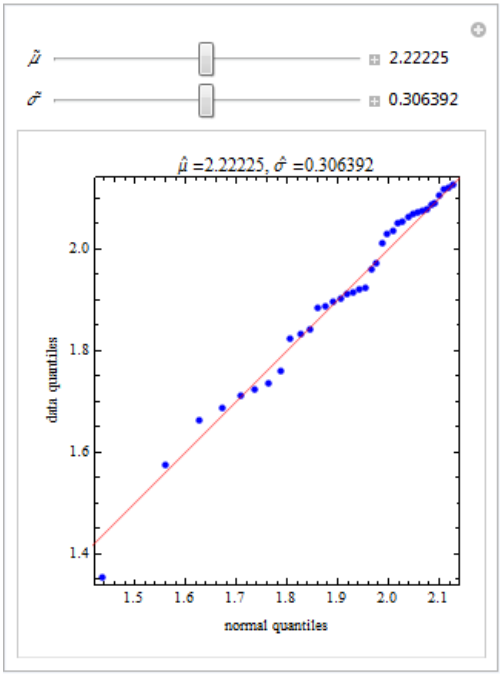

Figure 2.13: Locomotive data

value respectively, so in this case first refers to the data on and after March 24, 2005 while second refers to the data before this time point.

There is no statistically significant autocorrelation with the two parts of the series. So the two series may each be fit using the algorithm for the singly left-censored NID case.

A simple statistical model for the data is thus a level shift about white noise. For simplicity we allow a shift in the variance as well so, $z_t = \mu_t + a_t$, where

$$\mu_t = \begin{cases} \mu_1 & t \leq 96 \\ \mu_2 & t > 96 \end{cases}$$ and a_t is independent normally distributed with mean 0 and variance σ_t^2, where

$$\sigma_t^2 = \begin{cases} \sigma_1^2 & t \leq 96 \\ \sigma_2^2 & t > 96 \end{cases}$$

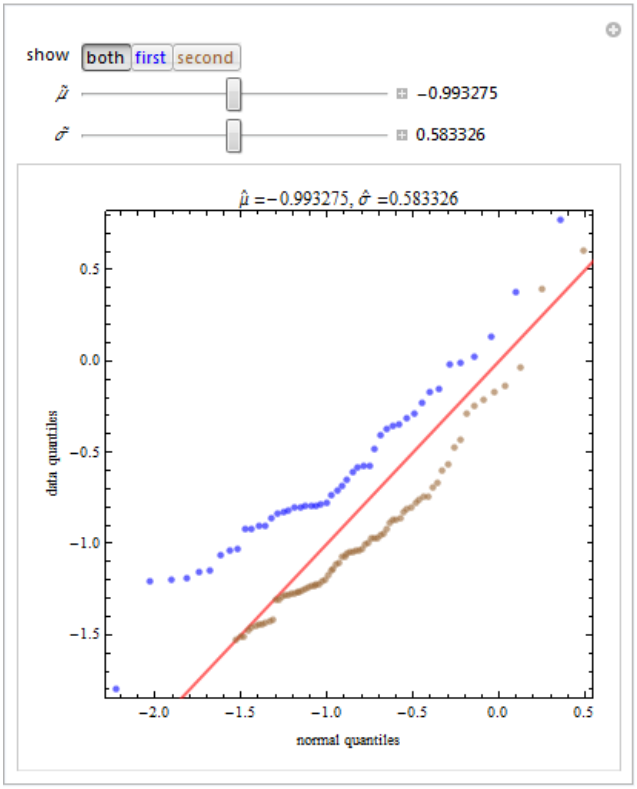

Figure 2.14: Dynamic normal plot for toxic water quality time series

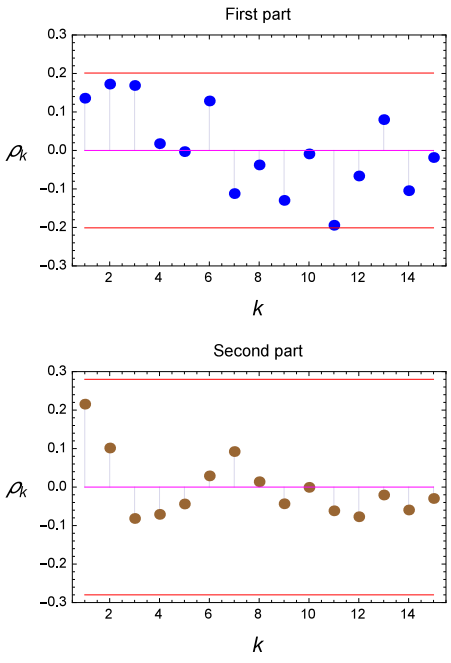

Figure 2.15: Autocorrelation functions.

Equivalently we may regard the data as being random samples from two independent normal distributions. The script and output for fitting this model using the CM algorithm is given in the display below.

```
> require("cents")
> z <- log(NiagaraToxic$toxic)
> iz <- c("o", "L")[1+NiagaraToxic$cQ]
> CM(z[1:96],iz[1:96])$est
            mle    se(mle)
mean -1.1499277 0.05773584
sd    0.5467298 0.04642786
> CM(z[-(1:96)],iz[-(1:96)])$est
            mle    se(mle)
mean -0.6819362 0.07623673
sd    0.5278323 0.05489638
```

The difference in the means is clearly highly significant but there does not appear to be any difference in the variances.

Chapter 3

Censored time series analysis

Missing values and censoring

In this chapter we will assume that the latent time series, $z_t, t = 1, \ldots, n$, is generated by a GLP as in Chapter 1. Usually this GLP will be an ARMA or stationary and invertible ARFIMA model but it could include other time series models such as the fractional Gaussian noise model. The observed series is determined by

$$y_t = \begin{cases} z_t & z_t > c_t^{(l)} \text{ and } z_t < c_t^{(u)} \\ c_t^{(l)} & y_t < c_t^{(l)} \\ c_t^{(u)} & y_t > c_t^{(u)} \end{cases} \tag{3.1}$$

where $c_t = \left(c_t^{(l)}, c_t^{(u)}\right)$ is the censoring process which may be stochastic or deterministic and must be independent of z_t and it is assumed to be known, as is the usual case in practice. Table 3.1 describes the type of censoring for the tth observation that are of interest.

$c_t^{(l)}$is finite	$c_t^{(u)} = \infty$	left censoring
$c_t^{(l)} = -\infty$	$c_t^{(u)}$is finite	right censoring
$c_t^{(l)}$is finite	$c_t^{(u)}$is finite	interval censoring
$c_t^{(l)} = \infty$	$c_t^{(u)}$any value	missing value
$c_t^{(l)}$any value	$c_t^{(u)} = -\infty$	missing value
$c_t^{(l)} = -\infty$	$c_t^{(u)} = \infty$	no censoring

Table 3.1: Censoring process.

In many cases, as in the air quality data mentioned in Chapter 1, the data are singly-censored which means that the censoring process is independent of t so the subscript t may be dropped. As well interval censoring is less frequently encountered. The most common censoring with actual water and air quality time series is left-censoring and not infrequently with only a single censor point.

We see that the missing value problem is closely related to the censoring problem. Consider the case of a single censor point for a left-censoring time series, denoted by $c^{(l)}$. As $c^{(l)}$

increases from $-\infty$, fewer and fewer observations are censored and so more and more information becomes available. So missing values represent the worst case in the larger censoring problem.

Thus our censoring algorithms should also be able to solve the missing value problem as well. Since the built-in R function arima() provides an exact MLE treatment using the Kalman filter algorithm, we can compare the estimates our new quasi-EM algorithm with those produced by the arima() function.

To illustrate the methods for dealing with missing values we consider a simple example with the latent series, z_t, that was generated using a particular starting value for the random number generator and $n = 50$, $\phi = 0.5$, $\mu = 100$, $\sigma_a = 5$. The simulated latent series is shown in the first panel in Figure 3.1. The observed series was generated by randomly deleting 25 data points, to obtain the y-series, shown in the second panel in Figure 3.1. The exact values are listed below the graphs.

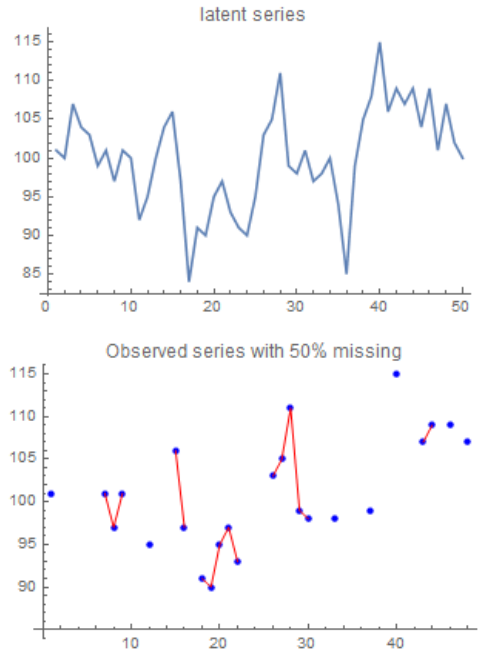

Figure 3.1: The simulated latent series and the observed series with 50% missing

For completeness the observed series, y_t, is listed in the display below,

```
101,NA,NA,NA,NA,NA,101,97,101,NA,NA,95,NA,NA,106,97,NA,91,90,
95,97,93,NA,NA,NA,103,105,111,99,98,NA,NA,98,NA,NA,NA,99,NA,NA,.............(2)
115,NA,NA,107,109,NA,109,NA,107,NA,100
```

and the latent series, z_t, is

```
101,100,107,104,103,99,101,97,101,100,92,95,100,104,106,97,84,91,90,
95,97,93,91,90,95,103,105,111,99,98,101,97,98,100,94,85,99,105,108,.........(3)
115,106,109,107,109,104,109,101,107,102,100
```

This example data is also available in our R package cents. The display below shows the AR(1) model that is fitted using the R function arima().

```
> arima(y, order=c(1,0,0))

Call:
arima(x = y, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.6334   101.2144
s.e.  0.1349     1.8703

sigma^2 estimated as 22.4:  log likelihood = -77.13,  aic = 160.25
```

Interpolation and missing values

The problem of fitting time series models to data with missing values is different problem from the interpolation problem. As pointed out by Shumway and Stoffer [2000], the Kalman filter algorithm provides a methods for both model fitting as well as for interpolation.

To highlight the difference between the interpolation problem and model estimation problem we provide a brief overview of the interpolation problem in this section.

Consider a stationary latent time series $z_t, t = 1, \ldots, n$ with mean μ and autocovariance function $\gamma_k = \text{Cov}(z_t, z_{t-k})$ and suppose that $t_1, \ldots, t_m$ denote the indices for the observed series and the complementary indices corresponding to the missing values are denoted by $s_1, \ldots, s_{n-m}$, Let $y = (z_{t_1}, \ldots, z_{t_m})'$ and $u = (z_{s_1}, \ldots, z_{s_{n-m}})'$ denote the vectors of latent series corresponding to the observed and missing values. Then the covariance matrix of y, denoted by Γ_y, is obtained by selecting the rows and columns corresponding to $t_1, \ldots, t_m$ from the $n \times n$ covariance matrix of $z_1, \ldots, z_n$, that is, $\Gamma_n = \left(\gamma_{i-j}\right)_{n\times n}$. Similarly the cross-covariance matrix, $\text{cov}(y, u) = \Gamma_{y,u}$ may be obtained by selecting rows corresponding to $t_1, \ldots, t_m$ and columns $s_1, \ldots, s_{n-m}$. Then the optimal interpolation may be obtained from the result for the conditional expectation in a multivariate normal distribution and is given by,

$$\mu + \Sigma_y^{-1}\Gamma_{y,u}(y - \mu) \tag{3.2}$$

where we have used the convention in R and *Mathematica*, that a vector minus a constant is defined by subtracting the constant from each element of the vector.

In the AR(1) case, $\gamma_k = \sigma_a^2\phi^k/\left(1 - \phi^2\right)$ and a simple result may be derived using *Mathematica* symbolics,

```
n = 11;
Gn = ToeplitzMatrix[Table[ϕ^k, {k, 0, n - 1}]] / (1 - ϕ^2);
i = Ceiling[n / 2];
C = Delete[Range[n], i];
Gc = Transpose[Gn[[C]]][[C]];
g = Delete[Gn[[6]], 6];
Gci = FullSimplify[Inverse[Gc]];
Z6 = Delete[Table[Subscript[z, t], {t, 1, 11}], 6];
FullSimplify[Gci.g].Z6
```

In[1]:=

Out[9]= $\frac{\phi\, z_5}{1+\phi^2} + \frac{\phi\, z_7}{1+\phi^2}$

This result differs from the usual forward forecast for z_6 given the past, ϕz_5, or the forecast for z_6 given the future, ϕz_7. In general if the tth observation is missing in an AR(1), the optimal interpolated value for z_t is given by $\phi\left(z_{t-1} + z_{t+1}\right) / \left(1 + \phi^2\right)$.

There is an extensive literature on the estimation of missing values or optimal interpolation in stationary and non-stationary time series. The univariate case is discussed using the Kalman filter by Kohn and Ansley [1986] and more generally for multivariate time series by Durbin and Koopman [2012]. Non-Kalman approaches to the time series interpolation problem are given by Nieto and Martinex [1996, 1991], Kasahara et al. [2009]. Their papers cite many other previous works on this topic.

Likelihood with missing values

Let $z_1, \ldots, z_n$ denote n successive values from an ARMA(p, q) model with mean μ, innovation variance σ_a^2 and coefficient parameters $\beta = \left(\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q\right)$. We assume that m of these values, $z_{t_1}, \ldots, z_{t_m}$, are observed and $n-m$ are missing and the mechanism by which the missing values were generated is unrelated to the underlying latent time series, $z_1, \ldots, z_n$. Let $t_i, i = 1, \ldots, m$, where $1 \le m \le n$, denote the time indices corresponding to the observed values so the actual observed time series, y_t, is

$$y_t = \begin{cases} z_t & t \in C \\ \text{NA} & t \notin C \end{cases}, \tag{3.3}$$

where NA indicates a missing value and C is the vector of indices for the complete observations, $C = \{t_1, \ldots, t_m\}$. Let $\gamma_k = \text{Cov}\left(z_t, z_{t-k}\right)$, $k = 1, \ldots, n$ be the autocovariance function for the latent time series, so the covariance matrix for the fully observed series $z_1, \ldots, z_n$ can be written $\Gamma_n = \sigma_a^2\left(\gamma_{i-j}\right)$, where the (i, j)-entry in the $n \times n$ matrix is indicated and where $\gamma_{-k} = \gamma_k$. We call $r = (n - m)/n$ the missing value rate.

The covariance matrix for $y_{t_1}, \ldots, y_{t_m}$ denoted by Γ_m is obtained by selecting rows and columns of Γ_n corresponding to $t_1, \ldots, t_m$. Then the exact log-likelihood for the parameters $\left(\mu, \beta, \sigma_a^2\right)$ including the constant term may be written,

$$\mathcal{L}\left(\mu, \beta, \sigma_a^2; y\right) = -\frac{m}{2}\log(2\pi) - \frac{m}{2}\log\sigma_a^2 - \frac{1}{2}\log\det(\Gamma_m) - \frac{1}{2\sigma_a^2}(y-\mu)'\Gamma_m^{-1}(y-\mu), \tag{3.4}$$

where $(y-\mu)' = (y_{t_1} - \mu, \ldots, y_{t_m} - \mu)'$ and $\det(\Gamma_m)$ denotes the determinant of Γ_m.

For MLE purposes we work with the profile or concentrated log-likelihood that is obtained by maximizing over σ_a^2 and dropping the constant term involving 2π. Then the profile log-likelihood function can be written,

$$\mathcal{L}(\mu, \beta; y) = -\frac{1}{2}\log\det(\Gamma_m) - \frac{m}{2}\log((y-\mu)'\Gamma_m^{-1}(y-\mu)/m), \tag{3.5}$$

and

$$\hat{\sigma}_a^2 = (y-\mu)'\,\Gamma_m^{-1}(y-\mu)\Big/ m \tag{3.6}$$

The direct approach to this problem is to form the exact likelihood function given in eqn.(3.5) and optimize it numerically. The only possible disadvantage of this approach is computational. The computational complexity as measured by the number of floating computations is $O\left(m^3\right)$ per function evaluation whereas other approaches reduce this to $O(m)$ or $O\left(m^2\right)$. This may be important when m is very large but it is much less of concern with modern computers. For example, taking $n = 3000$, $r = 0.5$, and $m = 1500$, the evaluation of $\mathcal{L}$ in eqn. (3.5) for an AR(1) model using simulated data takes less than 1 second using *Mathematica* on a Windows PC with a 2009 i7 CPU. So in many cases the direct evaluation of eqn. (3.5) is a very feasible approach. If programmed in an efficient computer language such as C, it would be a lot faster.

The Kalman filter provides the most widely used approach for fitting time series models with missing values. It provides a computationally efficient approach to the evaluation of the direct likelihood in eqns. (3.4) and (3.5) in the case of ARMA models. [Box et al., 2008, §13.3] review the Kalman filter algorithm for missing values for ARIMA models and provide a detailed overview for the AR(1) model. Jones [1980] provided the first treatment of fitting ARMA models with missing data using the Kalman filter and we provide a general algorithm for implementing the essential idea in Jones [1980] using general linear time series modelling algorithms presented in McLeod et al. [2007]. Our approach is more general since it is not obvious how the general linear process may be fit using the Kalman filter. Park et al. [1997] extended the Kalman filter missing value approach to the class of stationary ARFIMA models, so extensions are possible but further work with the Kalman filter methods is needed to explain how to extend this other more general linear processes such as fractional Gaussian noise as well as homogeneous non-stationary models. Another difficulty with the Kalman filter approach mentioned by some researchers is accuracy. Penzer and Shea [1997, p. 920] pointed out that many researchers have reported problems due to round-off error with the Kalman filter approach. The Kalman filter for ARMA models has computational complexity $O(n)$.

Ljung [1982] derived an efficient algorithm for the evaluation of the log-likelihood function in eqns. 3.4 and 3.5 that generalizes the earlier ARMA algorithm of Ljung and Box [1979] that was derived from the unconditional sum of squares function developed in celebrated time series book Box et al. [2008].

Another approach to the efficient evaluation of the exact likelihood given in eqns. 3.4 and (3.5) was obtained by Penzer and Shea [1997]. Their approach was derived by specializing the modified Cholesky decomposition algorithm given by Ansley [1979]. Penzer and Shea [1997] present computing timings that such their approach outperforms the Kalman filter method but I don't think this is reliable since the implementation details are complicated and such raw

computer timings depend very much on the implementation details. Also with current computing capabilities, computational speed is often not of much practical importance in most actual applications since n is usually less not very large. However larger n may occur with some financial time series and Big Data is a current area of development so computationally efficient methods may be needed in some cases.

New computationally efficient algorithm for fitting AR(1) in complete data case

In this section no missing values are assumed so we work directly with the latent series $z_t, t = 1, \ldots, n$. This algorithm introduces notation and methods that will be used when we discussed the censored case. This algorithm is important because as we will demonstrate later in Figure 3.4 the output from R's arima() function is unreliable even in the AR(1) case. This makes the arima() function problematic suitable for computationally intensive statistical inference methods such as bootstrapping.

The new algorithm, called FullMLEAR1 is based on the exact MLE algorithm for the AR(1) described in Ying Zhang et al. [2013] and implemented in R in the package mleur on CRAN and our improvement is to provide an efficient computation of exact MLE for both parameters ϕ and μ. The algorithm discussed in Ying Zhang et al. [2013] is exact in the case of known mean. In practice, the series is simply corrected for the sample mean and the mean corrected series is treated as a mean zero time series. Since the sample mean is asymptotically efficient, this method works well in practice in many situations. However the built-in R function arima() uses the exact MLE estimate for the sample mean, so it was decided for the purposes of comparison to modify the algorithm of Ying Zhang et al. [2013] to provide exact MLE estimates for (μ, ϕ).

From the result given in McLeod et al. [2007, eqn. 9] the concentrated log-likelihood function for the latent series is,

$$\mathcal{L}(\mu, \phi) = -\frac{n}{2}\log(S/n) - \frac{n}{2}\log\left(1-\phi^2\right) \tag{3.7}$$

where,

$$S = (z_1-\mu)^2\left(1-\phi^2\right) + \sum_{t=2}^{n}\left(z_t - \mu - \phi\left(z_{t-1}-\mu\right)\right)^2 \tag{3.8}$$

Also $\hat{\sigma}_a^2 = S/n$. The new algorithm is derived by solving $\partial_\mu S = 0$, where S is the sum-of-squares function given in eqn. (3.8). Hence,

$$\partial_\mu\left(\left(z_1-\mu\right)^2\left(1-\phi^2\right) + \textstyle\sum_{t=2}^{n}\left(z_t-\mu-\phi\left(z_{t-1}-\mu\right)\right)^2\right)$$

$$= -2\left(z_1-\mu\right)\left(1-\phi^2\right) + \textstyle\sum_{t=2}^{n} 2\left(z_t-\mu-\phi\left(z_{t-1}-\mu\right)\right)(-1+\phi)$$

$$= -2\left(z_1-\mu\right)\left(1-\phi^2\right) - 2\textstyle\sum_{t=2}^{n}(1-\phi)\left(z_t-\phi z_{t-1}-\mu(1-\phi)\right)$$

Next we set $\partial_\mu S = 0$ and solve for μ.

$$
\begin{aligned}
-2(z_1-\mu)\left(1-\phi^2\right)-2\sum_{t=2}^{n}(1-\phi)\left(z_t-\phi z_{t-1}-\mu(1-\phi)\right)&=0\\
\Leftrightarrow (z_1-\mu)\left(1-\phi^2\right)+\sum_{t=2}^{n}(1-\phi)\left(z_t-\phi z_{t-1}-\mu(1-\phi)\right)&=0\\
\Leftrightarrow (z_1-\mu)\left(1-\phi^2\right)+\sum_{t=2}^{n}(1-\phi)\left(z_t-\phi z_{t-1}\right)-\sum_{t=2}^{n}\mu(1-\phi)^2&=0\\
\Leftrightarrow (z_1-\mu)\left(1-\phi^2\right)-(n-1)\mu(1-\phi)^2+(1-\phi)\sum_{t=2}^{n}\left(z_t-\phi z_{t-1}\right)&=0\\
\Leftrightarrow z_1\left(1-\phi^2\right)-\mu\left(1-\phi^2\right)-(n-1)\mu(1-\phi)^2+(1-\phi)\sum_{t=2}^{n}\left(z_t-\phi z_{t-1}\right)&=0\\
\Leftrightarrow z_1\left(1-\phi^2\right)+(1-\phi)\sum_{t=2}^{n}\left(z_t-\phi z_{t-1}\right)-\mu\left(1-\phi^2\right)-(n-1)\mu(1-\phi)^2&=0\\
\Leftrightarrow z_1\left(1-\phi^2\right)+(1-\phi)\sum_{t=2}^{n}\left(z_t-\phi z_{t-1}\right)=\mu\left(\left(1-\phi^2\right)+(n-1)(1-\phi)^2\right)&
\end{aligned}
$$

Hence,

$$\hat{\mu}=\frac{z_1\left(1-\phi^2\right)+(1-\phi)(A-\phi B)}{1-\phi^2+(n-1)(1-\phi)^2} \tag{3.9}$$

where,

$$A=\sum_{t=2}^{n}z_t,\ B=\sum_{t=1}^{n-1}z_{t-1}$$

Iterative exact MLE algorithm. FullMLEAR1.

Step 1. Set $\hat{\mu}_0=\bar{z}$, where $\bar{z}=n^{-1}\sum_t z_t$ and $i=0$.

Step 2. Set $u_t=z_t-\hat{\mu}_i$ and use the exact MLE algorithm of Zhang, Yu and McLeod (2013) to fit the AR(1) model to u_t and obtain $\hat{\phi}^{(i)}$.

Step 3. Using eqn. (3.9) obtain, $\hat{\mu}^{(i)}$.

Step 4. Evaluate the log-likelihood function using eqn. (3.7).

Step 5. Stop if the log-likelihood function has converged. Otherwise, set $i\rightarrow i+1$ and repeat Steps 2, 3, and 4.

R code snippet

The **FullMLEAR1** algorithm is implemented in our R package cents in the function fitar1(). The code snippet below compares the output of our function with the built-in R function arima() for the simulated latent time series. Our fitar1() function returns a vector with $\left(\hat{\mu},\hat{\phi},\mathcal{L}\left(\hat{\mu},\hat{\phi}\right)\right)$. The agreement with the parameter estimates is satisfactory. The values of the log-likelihood differ because we use the profile log-likelihood.

```
> fitar1(z)
[1] 100.1804805    0.6647272  -95.5147693
> arima(z, order=c(1,0,0))

Call:
arima(x = z, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.6648  100.1808
s.e.  0.1079     2.1219

sigma^2 estimated as 26.9:  log likelihood = -153.54,  aic = 313.08
```

An EM algorithm for missing values in AR(1)

The idea with for the Kalman filtering algorithm is to replace any missing values with their forward predictions and then evaluate the log-likelihood function. Previously Shumway and Stoffer [2000] have elucidated the connection between the Kalman filter algorithm for missing values and the EM algorithm. In this section, we present an alternative approach.

A new exact algorithm for computing direct log-likelihood given in eqn. (3.16) is developed. This algorithm is basically equivalent to the algorithm used in the Kalman filter approach but the implementation details are completely different. Although the approach in this section is just developed for the AR(1) model it could easily be extended to the more general ARMA case.

Given the latent series $z_1, \ldots, z_n$ and define y_t and C as in eqn. (3.3). The observed values, $y_t, t = t_1, \ldots, t_m$ and the missing values correspond to the subseries $z_{s_1}, \ldots, z_{s_{n-m}}$, where $s_1 < s_2 < \ldots < s_{n-m}$ are the indices corresponding to each of the missing values. For any $t \geq 1$, set

$$\mathcal{Z}_t = (z_t, z_{t-1}, \ldots, z_1)' \tag{3.10}$$

and set

$$\mathcal{U} = (z_{s_1}, \ldots, z_{s_{n-m}})' . \tag{3.11}$$

We assume without loss of generality that $t_1 = 1$ and $t_m = m$. The model parameters are $\lambda = \left(\mu, \phi, \sigma_a^2\right)$. The joint density corresponding to the observed values, $y_t, t = t_1, \ldots, t_m$ can be written,

$$f\left(\mathcal{Z}_n; \lambda\right) = f\left(z_{t_1}; \lambda\right) f\left(z_{t_2} | \mathcal{Z}_{t_2-1}; \lambda\right) f\left(z_{t_3} | \mathcal{Z}_{t_3-1}; \lambda\right) \ \ldots f\left(z_{t_m} | \mathcal{Z}_{t_m-1}; \lambda\right) f(\mathcal{U}; \lambda). \tag{3.12}$$

Substituting, $y_{t_i} = z_{t_i}, i = 1, \ldots, m$ and taking logarithms,

$$\log f\left(\mathcal{Z}_n; \lambda\right) = \log f\left(y_1; \lambda\right) + \sum_{i=2}^{m} \log f\left(y_{t_i} | \mathcal{Z}_{t_i-1}; \lambda\right) + \log f(\mathcal{U}; \lambda) \tag{3.13}$$

We see that this is exactly the form required for the EM algorithm with objective function,

$$Q(\lambda'|\lambda) = \log f(y_1;\lambda') + \sum_{i=2}^{m} \log f\left(y_{t_i}|\mathrm{E}_\lambda\left\{\mathcal{Z}_{t_i-1}\right\};\lambda'\right), \tag{3.14}$$

where E_λ is expectation with respect to λ. The EM algorithm iterations are specified by,

$$\lambda^{(i)} = \underset{\lambda'}{\operatorname{argmax}} Q\left(\lambda'|\lambda^{(i-1)}\right) \tag{3.15}$$

Note that eqn. (3.14) holds for ARMA models as well if we make change the notation so that λ includes all the parameters, so the algorithm may be implemented for this more general case. In the AR(1) case we proceed by defining for $t = 1$,

$$[w_1] = \begin{cases} y_t, & t \in C, t \geq 2, \\ \mu, & t \notin C, t \geq 2, \end{cases} \tag{3.16}$$

and for $t = 2, \ldots, n$,

$$[w_t] = \begin{cases} y_t, & t \in C, \\ \mu + \phi\left([w_{t-1}] - \mu\right), & t \notin C. \end{cases} \tag{3.17}$$

So if $t, t-1, \in C$, $f(z_t|\mathcal{Z}_{t-1})$, where $f(z_t|\mathcal{Z}_{t-1})$ is the normal PDF with mean $\mu + \phi(z_{t-1} - \mu)$ and variance σ_a^2. More generally for any $t \in C$, $f([w_t]\,|\,[w_{t-1}]) = f(z_t|\mathcal{Z}_{t-1})$, where $f(z_t|\,[w_{t-1}])$ is normally distributed with mean $\mu + \phi([w_{t-1}] - \mu)$ but the variance is more complicated. In general, assume that $t-1, t-2, \ldots, t-k-1 \notin C$ and $t-k \in C$. So the lag separation between y_t and the closest previous observation is k lags. Then $[w_{t-1}]$ as recursively defined in eqn. (3.16) is equivalent to the forecast for y_t from lead time y_{t-k} and hence has variance $v_k = \left(1 + \phi^2 + \ldots + \phi^{2(k-1)}\right)\sigma_a^2$. The variances may be recursively computed along with $[w_t]$. Set $v_1 = \sigma_a^2/\left(1-\phi^2\right)$ and for $t = 2, \ldots, n$ set,

$$v_t = \begin{cases} \sigma_a^2, & t \in C, \\ 1 + \phi^2 v_{t-1}, & t \notin C. \end{cases} \tag{3.18}$$

Letting $\varphi\left(z;\mu,\sigma^2\right)$ denote the normal distribution with mean μ and variance parameter σ^2, the exact log-likelihood function for $y_{t_1}, \ldots, y_{t_m}$ may be written,

$$\mathcal{L}\left(\mu,\phi,\sigma^2|y_{t_1},\ldots,y_{t_m}\right) = \varphi\left([w_1];\mu,\phi,\sigma_a^2/\left(1-\phi^2\right)\right) + \sum_{t=2}^{m} \varphi\left([w_t];\mu+\phi\left([w_{t-1}]-\mu\right),\phi,v_t\right) \tag{3.19}$$

As a check, a *Mathematica* package was implemented to compute the log-likelihood for an AR(1) using eqn.(3.19) as well as with the direct log-likelihood function based on eqn.(3.16). Both of these *Mathematica* functions gave results on the test data discussed in (2) identical to that produced by the Kalman filter algorithm that is implemented in the R function arima() and whose output is displayed below the data (2).

The approach in this section will now be modified and simplified. A convenient algorithm for the more general censored case.

Quasi-EM algorithm

If we are given the latent series, $z_1, \ldots, z_n$ then the joint probability density function can be written,

$$f(z_1, \ldots, z_n) = f(z_1) f(z_2|z_1) f(z_3|z_2) \ \ldots f(z_n|z_{n-1}) \tag{3.20}$$

For $t = 2, \ldots, n$, the conditional distribution $f(z_t|z_{t-1})$ is normal with mean $\mu_t = \mu + \phi(z_{t-1} - \mu)$ and variance σ_a^2 and for $t = 1$, z_1 is normal with mean μ and variance $\sigma_a^2/\left(1 - \phi^2\right)$. Hence the concentrated log-likelihood may be computed. From the result given in McLeod et al. [2007, eqn. 9],

$$\mathcal{L}(\mu, \phi) = -\frac{n}{2} \log(S/n) - \frac{n}{2} \log\left(1 - \phi^2\right) \tag{3.21}$$

where,

$$S = (z_1 - \mu)^2 \left(1 - \phi^2\right) + \sum_{t=2}^{n} (z_t - \mu - \phi(z_{t-1} - \mu))^2 \tag{3.22}$$

Also $\hat{\sigma}_a^2 = S/n$.

To deal with the missing values, we set $[y_t]$ equal to its conditional expectation given the observations up to and including time t, so for $t = 2, \ldots, n$,

$$[y_t] = \begin{cases} y_t, & t \in C, \\ \mu + \phi([y_{t-1}] - \mu), & y_t \notin C, \end{cases} \tag{3.23}$$

For $t = 1$, if $y_1 = \text{NA}$ then $[y_1] = \mu$ otherwise $y_1 = z_1$.

In the EM approach, we set $\beta' = (\mu', \phi')$ and $\beta = (\mu, \phi)$. Here the β' are the parameters that we maximize over and β are the parameters used in the expectation step.

$Q(\beta'|\beta) = -\frac{n}{2} \log(S/n) - \frac{n}{2} \log\left(1 - (\phi')^2\right)$

$$S = ([w_1] - \mu')^2 \left(1 - (\phi')^2\right) + \sum_{t=2}^{n} ([w_t] - \mu' - \phi'([w_{t-1}] - \mu'))^2 \tag{3.24}$$

$$[w_t] = \begin{cases} y_t, & t \in C, t \geq 2, \\ \mu + \phi([w_{t-1}] - \mu), & t \notin C, t \geq 2 \end{cases} \tag{3.25}$$

For $t = 1$, if $[w_1] = \text{NA}$ then $[w_1] = \mu$ otherwise $w_1 = z_1$. Then the (quasi-MLE) QMLE are obtained by the iterative process EM steps.

Innovation variance. From eqn. (3.4) an estimate of the innovation variance is given by,

$$\hat{\sigma}_a^2 = \frac{1}{m} \sum_{t \in C} ([w_t] - \mu' - \phi'([w_{t-1}] - \mu'))^2 \tag{3.26}$$

An alternative approach would be to use the full likelihood for $\left(\mu, \phi, \sigma_a^2\right)$ as defined in eqn. (3.20) to obtain the exact MLE for all three parameters.

Quasi-EM algorithm for missing values in AR(1). QMLEAR1.
Step 0. Select initial estimate β.
Step 1. Expectation. Compute the expectations using eqn. (3.16).
Step 2. Maximization.

$$\hat{\beta} = \underset{\beta'}{\operatorname{argmax}} Q(\beta'|\beta) = -\frac{n}{2}\log(S/n) - \frac{n}{2}\log\left(1-(\phi')^2\right) \tag{3.27}$$

Step 3. Stop if the estimates have converged otherwise replace β with $\hat{\beta}$ return to Step 1.

The maximization step in eqn. (3.27) can be efficiently evaluated by using the algorithm **FullMLEAR1** algorithm. The **QMLEAR1** algorithm is implemented in our cents R package in the function fitcar1().

The Figure below compares the QMLEAR1 estimates for (μ, ϕ) using the data (2) with the exact MLE computed by the direct method using eqn. (3.16). The magenta, red and black dots represent the QMLEAR1, direct MLE, and true parameter values respectively. The contours shown the 50%, 90% and 95% confidence regions computed using the likelihood ratio test method with the likelihood defined in eqn. (3.16). We see that the difference between the QMLEAR1 and exact MLE is negligible.

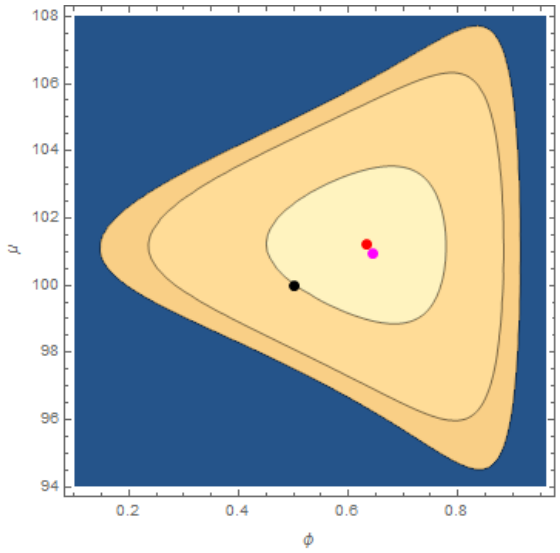

Figure 3.2: Comparison of algorithms for estimating MLE in the simulated example.

Simulation comparisons

Normal scenarios

Our normal scenarios cover missing value AR(1) model with parameters and missing values rates 20% and 50%. In the next section we investigate more extreme cases involving 90% missing values. Some simulations were done to compare our proposed algorithm fitcar1()with the built-in R function arima().

For series lengths $n = 50, 100, 200, 500$, parameters $\phi = -0.9, -0.6, -0.3, 0.0, 0.3, 0.6, 0.9$, $\mu = 0$ and $\sigma_a = 1$, and missing values rates 20% and 50%, 1000 simulations were done for each setting. The plots of the root-mean-square error (RMSE) shown below shown that in most cases there is little difference between these two algorithm except that in some cases fitcar1() is slightly more accurate.

Figure 3.3 compares the RMSE for the estimates of μ. It is interesting that when $\phi < 0$, the RMSE seems to increase slightly with n and the reverse is true when $\phi \geq 0$. This must be a finite sample effect and due to the fact that the estimation of the parameter is greatly improved when there is negative autocorrelation. A similar effect occurs in the principle of anti-thetic sampling where a negative correlation is used to improve the estimation. The blue circles show the arima() estimates and the red dots are the fitcar1() estimates.

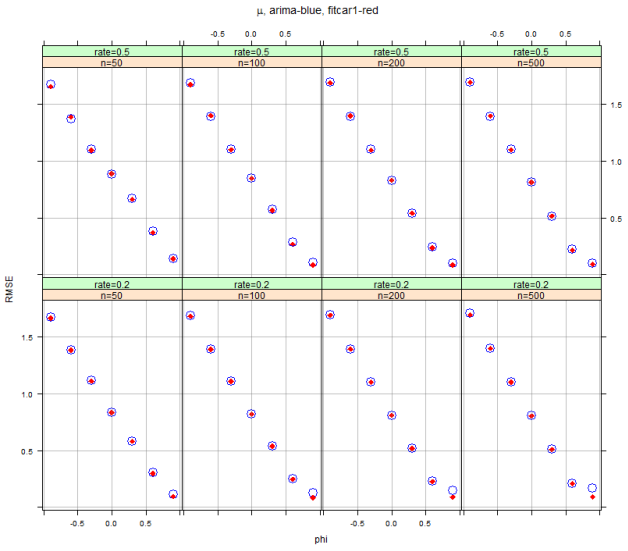

Figure 3.3: Comparing RMSE for estimation of the mean.

Figure 3.4 compares the RMSE for the estimates of ϕ. In this case the accuracy increases with n for all ϕ. When $n = 50$ and $\phi = 0.9$, the plots show that the RMSE greatly increases and the fitcar1() is noticeably more accurate.

Let $\hat{\phi}_i, i = 1, \ldots, 1000$ denote the estimate of ϕ in each of the 1000 simulations for a fixed setting. We compare the biases, $\phi - \hat{\phi}_i$ in Figure 3.5 for $\phi = 0.9$, $n = 50$, and 20% and 50% missing value rates. The biases greatly increases with the amount of missing data in the arima() case and less so in the fitcar1() case. The R function fitcar1() estimates also have lower biases than the arima() as we might expect from Figure 3.5. It is possible that the poor performance in the arima() case could be due to optimization algorithm.

More extreme missing values

Although extreme missing values rates of over 80% don't seem to arise in many applications, it might be that the approximations used in the **QMLEAR1** algorithm break down in this situation. So the simulation was repeated comparing **QMLEAR1** and arima() with missing value rates 80% and 90%. The results shown in Figures 3.6 and 3.7 confirm that there is little

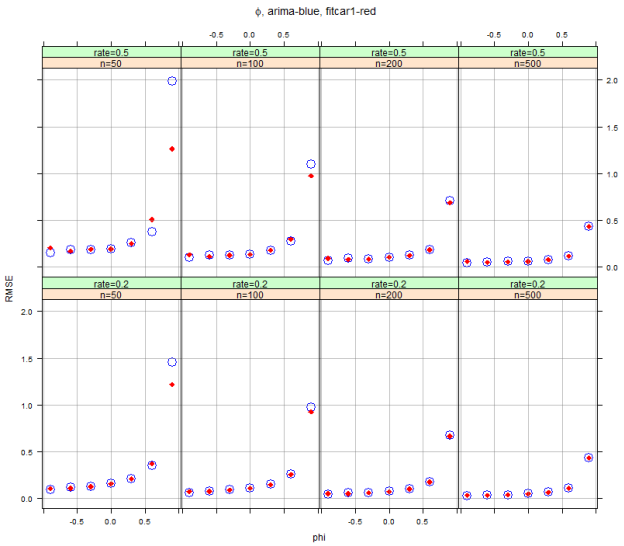

Figure 3.4: Comparing RMSE for estimation of the ϕ.

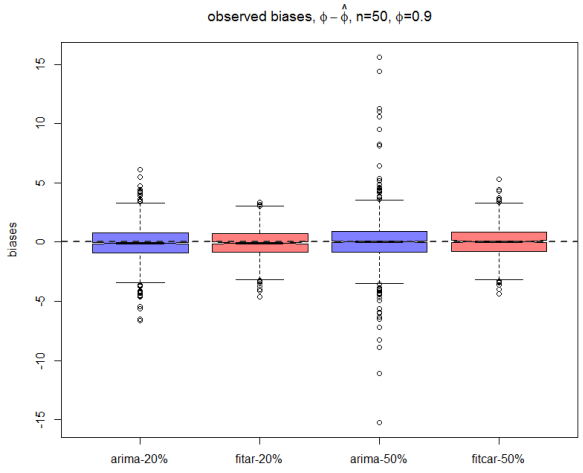

Figure 3.5: Boxplots comparing the biases for the estimates for ϕ using arima() and fitar()with 20 and 50 percent missing values.

difference between the algorithms in most cases and where there is a big difference such as in the estimate of ϕ when $\phi = 0.9$ and the series length is not long, the **QMLEAR1** algorithm produces a more accurate result.

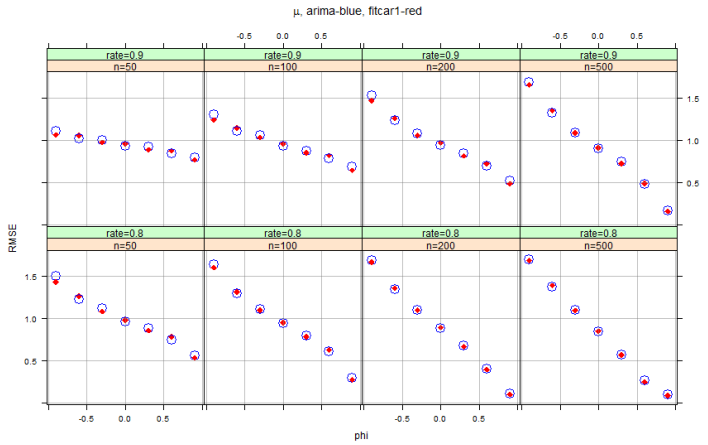

Figure 3.6: Comparing RMSE for estimation of the μ with missing value rates 80% and 90%

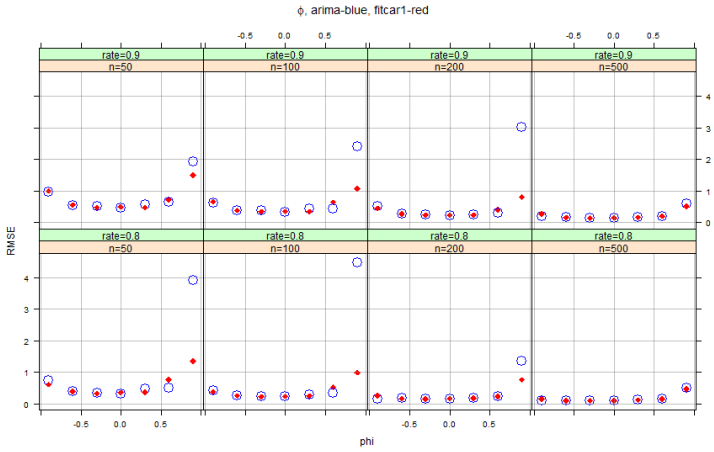

Figure 3.7: Comparing RMSE for estimation of the ϕ with missing value rates 80% and 90%

Concluding remarks

The missing value problem in time series model estimation may be regarded as an extreme case of censoring. As the censoring threshold in left-censoring is increased less and less information is available for model estimation and in the extreme case where the threshold is very large missing values are generated that contain no information. The fact that the **QMLEAR1** performs just as well as the exact approach using arima() for the AR(1) suggests the idea behind this algorithm will be useful in the more general censored case.

Censored AR(1) time series analysis

The full CENAR(1) for an observed time series includes a censoring process defined in eqn. (3.1) with the latent time series, z_t, defined by an AR(1) model, $z_t = \mu + \phi(z_{t-1} - \mu) + a_t$ where $a_t \sim \text{NID}\left(0, \sigma_a^2\right)$. As noted above in eqns. (3.7) and (3.8), which we repeat here for convenience, the concentrated log-likelihood for the latent series may be written,

$$\mathcal{L}(\mu, \phi) = -\frac{n}{2}\log(S/n) - \frac{n}{2}\log\left(1 - \phi^2\right) \tag{3.28}$$

where,

$$S = (z_1 - \mu)^2\left(1 - \phi^2\right) + \sum_{t=2}^{n}\left(z_t - \mu - \phi(z_{t-1} - \mu)\right)^2 \tag{3.29}$$

and $\hat{\sigma}_a^2 = S/n$. To deal with censoring, we replace the censored values by their conditional expected values given the past data, $\mathcal{Z}_{t-1}$, where $\mathcal{Z}_t$ is defined as in eqn. (3.10) and the censoring process, c_t, in Table 3.1 so

$$[w_t] = \begin{cases} y_t, & t \in C, \\ \mathrm{E}_{t,c}\{y_t\}, & y_t \notin C, \end{cases} \tag{3.30}$$

where $\mathrm{E}_{t,c}\{y_t\}$ is the conditional expectation given the past observations, $\mathcal{Y}_{t-1} = (y_{t-1}, y_{t-2}, \ldots, y_1)'$ and the censoring c_t. Assuming normality, $\mathrm{E}_{t,c}\{y_t\}$ is computed as the mean in a truncated normal distribution with truncation determined by c_t and normal mean parameter, μ_t,

$$\mu_t = \mu + \phi([w_{t-1}] - \mu) \tag{3.31}$$

and variance parameter, σ_a^2. Treating the sequence $[w_t], t = 1, \ldots, n$ as an AR(1) we obtain estimates for the parameters using the exact MLE algorithm, **FullMLEAR1**.

Initial estimates for μ, ϕ, and σ_a^2 may be obtained using the FullMLEAR1 algorithm on the observed data $y_t, t = 1, \ldots, n$ by ignoring the censoring, that is, by treating the censored values as fully observed value.

MLECAR1:Algorithm for censoring values in AR(1).
Step 0. Select initial estimate β.
Step 1. Expectation. Compute $[w_t], t = 1, \ldots, n$.
Step 2. Maximization. Fit using **FullMLEAR1**.
Step 3. Stop if the log-likelihood has converged.
This algorithm is implemented in our cents R package in the function fitcar1().

Comment on the exact likelihood function for CENAR(1)

Park et al. [2007] claim to derive the exact log-likelihood for the AR(1) in the singly censored case but from our work in the missing value case, we realize that their result is wrong. Recall that for the missing value problem in the AR(1) we found the likelihood function is not Markovian in the sense that $f(z_t|\mathcal{Z}_{t-1}) \neq f(z_t|z_{t-1})$ when $z_{t-1} \notin C$. Similarly, excluding the trivial cases where the censor limit is at $\pm\infty$, it is obvious that when there is non-trivial censoring

the resulting time series y_t is not Markovian even when the underlying latent process z_t is a stationary AR(1) and is thus Markovian.

As a simple check on the above remarks we simulated an AR(1), $z_t = \phi z_{t-1} + a_t, t = 1, \ldots, n$ with $\phi = 0.9$, $a_t \sim$ NID(0,1), and $n = 10^5$. Then we generated a censored series $y_t = \max(y_t, -1)$. For this series we found that the empirical estimates of D

$$\hat{p}_1 = \text{est}\Pr\{z_t \in C | z_{t-1} \notin C\} = 0.196247. \tag{3.32}$$

Next we generated another independent replication and this time computed,

$$\hat{p}_2 = \text{est}\Pr\{z_t \in C | z_{t-1} \notin C, z_{t-2} \notin C\} = 0.161406 \tag{3.33}$$

Applying the standard test for equality of binomial proportions yielded a Z-score of -10.9 so the proportions are certainly different as we claimed. As a check, the tests were repeated using $\phi = 0$ and we found $p_1 = 0.8493$, $p_2 = 0.8452$, and $Z = -0.52$. So as expected, since the censored time series y_t is equivalent to the random sampling situation discussed in Chapter 2, the proportions must be the same.

Of course the likelihood derived by Park et al. [2007] for the CENAR(1) may provide an approximation but the likelihood itself is quite complicated and the generalization to higher order AR models would be very complex.

Standard errors and inference on the estimates

The standard errors for the parameter estimates of μ and σ_a^2 may be obtained using bootstrapping. Both the parametric and non-parametric block bootstrap can be used.

The Figure 3.8 compares the estimated standard error for μ, $\hat{\sigma}_\mu$, using the block with block-length 10 and parametric bootstrap in simulated CENAR(1) models with series length $n = 200$, mean zero, $\phi = -0.9, -0.6, \ldots, 0.9$, unit innovation variance and left censoring rates $r = 0.2, 0.5$. For each setting, 2500 simulations were done. In each simulation $B = 1000$ bootstrap iterations were done. The blue dots show the average estimated standard deviation for $\hat{\mu}$. The estimated standard deviations times two are less than the width of the plotting symbol. These standard deviations are given in the Table 3.2

$\phi \backslash r$	0.2	0.5
-0.9	{0.067, 0.268 }	{0.298, 0.293}
-0.6	{0.050, 0.077 }	{0.171, 0.087}
-0.3	{0.057, 0.058 }	{0.212, 0.063}
0.0	{0.076, 0.060 }	{0.159, 0.060}
0.3	{0.108, 0.077 }	{0.128, 0.071}
0.6	{0.176, 0.126 }	{0.221, 0.109}
0.9	{0.314, 0.483 }	{0.379, 0.403}

Table 3.2: The estimated standard errors for estimated μ. The first entry in each pair is for the block bootstrap and the second for the parametric bootstrap

In an AR(1) with no censoring and n consecutive observations, the asymptotic variance of the MLE $\hat{\mu}$ is given by $\sigma_\mu^2(\phi, n) = \sigma_a^2 / \left(n(1-\phi)^2\right)$. Since $\sigma_\mu^2(\phi, n)$ ignores the effect of

censoring, we would expect $\sigma_\mu^2(\phi, n) < \sigma_\mu^2(\phi, c, n)$ where $\sigma_\mu^2(\phi, c, n)$ is the variance of the MLE for μ is the CENAR(1) model that is estimated in our simulations using the bootstraps.

Another bound for the of $\sigma_\mu^2(\phi, c, n)$ may be obtained using the expected information matrix in the previous chapter to obtain $\sigma_\mu^2(c, n)$ the variance of the mean in censored normal random samples of size n and censor point c. Since this ignores the autocorrelation effect we would expect $\sigma_\mu^2(c, n) < \sigma_\mu^2(\phi, c, n)$.

The red curves on the plot show the approximate estimates for σ_μ based on these two asymptotic formulas.

Overall the parametric bootstrap may be preferred. When $\phi = 0.6, 0.9$ and $r = 0.5$ both methods indicate the estimate of σ_μ has a much larger value than may be expected by the crude asymptotic effective sample size considerations. The block bootstrap appear less effective when $r = 0.5$ than the parametric bootstrap. Perhaps for $\phi > 0$ a larger block length needed.

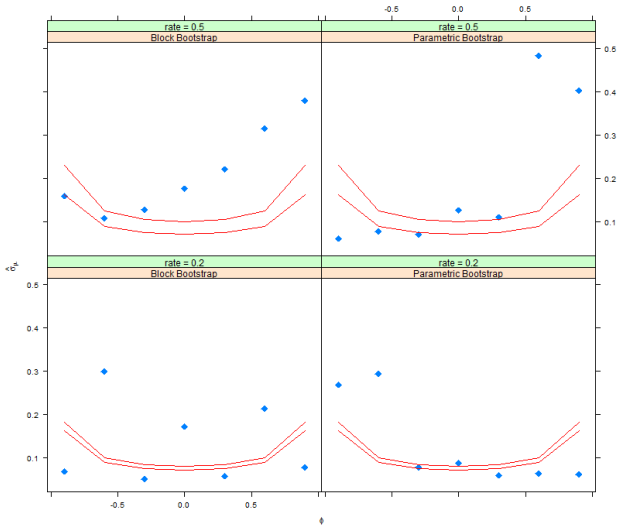

Figure 3.8: Compares the estimated standard error for $\mu, \hat{\sigma}_\mu$, using the block with block-length 10 and parametric bootstrap in simulated CENAR(1) models with series length $n = 200$, mean zero, $\phi = -0.9, -0.6, \ldots, 0.9$, unit innovation variance and left censoring rates $r = 0.2$, and $r = 0.5$

Model diagnostic check

The residuals may be defined by $\hat{a}_t = [y_t] - \hat{\phi}\,[y_{t-1}]$, $t = 2, \ldots, n$. The model adequacy may be assessed by using the Box-Pierce portmanteau diagnostic check,

$$Q_m = n \sum_{k=1}^{m} r_{\hat{a}}^2(k), \tag{3.34}$$

where

$$r_{\hat{a}}(k) = \left(\sum_{t=k+1}^{n} \hat{a}_t \hat{a}_{t-k} \right) / \left(\sum_{t=1}^{n} \hat{a}_t^2 \right). \tag{3.35}$$

The p-values for this diagnostic test may be computed using the Monte-Carlo test method discussed by Lin (2007) and Mahdi (2011). This algorithm is summarized below.

Algorithm MCTest for CENAR(1) Model

Step 0. Select M typically $M = 15$ but small or larger values may be used depending on whether or not higher order autocorrelations may be important if seasonality or periodicity is anticipated. Then for $m = 1, \ldots, M$, use the observed residuals from the fitted model to compute $Q_m^{(\text{obs})}$ for $m = 1, \ldots, M$. Select the number of iterations, say K. Usually $K = 250$ is adequate but $K = 1000$ may be preferable. Set the counter $i \longleftarrow 1$.

Step 1. Simulate the fitted censored time series, $y_t^{(i)}$ and then fit this model to obtain the residuals and the portmanteau statistics, $Q_m^{(i)}$, $m = 1, \ldots, M$,

Step 2. Increment the counter k_m if $Q_m^{(i)} > Q_m^{(\text{obs})}$.

Step 3. Increment the counter $i \longleftarrow i + 1$ and return to Step 1 and 2 if $i \leq K$.

Step 4. Plot the p-values, $p_m = (k_m + 1)/(K + 1)$, $m = 1, \ldots, M$. Low p-values provide evidence against the adequacy of the order 1 model and suggest that possibly a higher order model is needed.

Example

We illustrate with a simulated CENAR(1) model. The latent process is an AR(1) with mean $\mu = 100$ and innovation standard deviation $\sigma = 15$ and autocorrelation parameter $\phi = 0.8$. Then the CENAR(1) series is obtained by left-truncating so that about 50% of the data is truncated, so the truncation point is at $c = \mu - 15\Phi^{-1}(r) = 100$. A time series plot of a realization of length $n = 200$ is shown in Figure 3.9. Using our CM algorithm we obtained $\hat{\mu} = 101.05$ and $\hat{\sigma}_z = 23.74$. Note that the theoretical variance of the latent time series is $15 \Big/ \sqrt{\left(1 - 0.8^2\right)} = 25$. So both estimates are reasonably accurate. The CENAR(1) algorithm produces MLE $\hat{\phi} = 0.7107$ and $\hat{\sigma}_a = 12.52$.

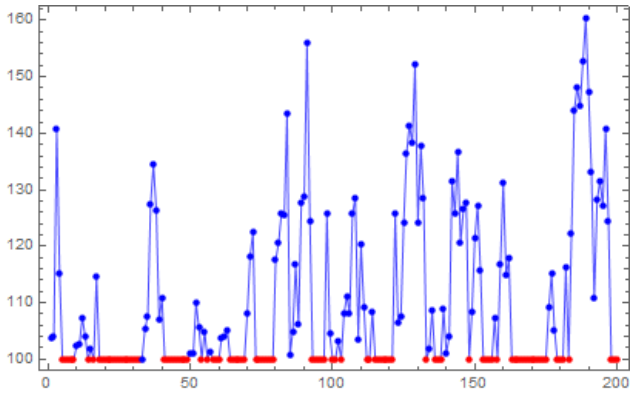

Figure 3.9: Time series plot of simulated CENAR(1)

Censoring algorithms for linear time series

Recursive computation of the inverse covariance matrix

Theorem: Γ_k^{-1} for $k = 1, 2, \ldots$ recursively setting, $\Gamma_1^{-1} = \left(\gamma_0^{-1}\right)$ and

$$\Gamma_{k+1}^{-1} = \begin{pmatrix} \Gamma_k^{-1}\left(\mathcal{I}_k + eaa'\right) & f \\ f' & e \end{pmatrix} \tag{3.36}$$

where $a = \Gamma_k^{-1}h$, $h = (\gamma_1, \ldots, \gamma_k)'$, $e = \left(\gamma_0 - h'\Gamma_k^{-1}h\right)^{-1}$, $f = -e\Gamma_k^{-1}h$ and $\mathcal{I}_k$ is the $k \times k$ identity matrix.

Proof:

The recursion is derived from the well-known theorem on the inverse of a partitioned matrix,

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} + A^{-1}B\left(D - CA^{-1}B\right)^{-1}CA^{-1} & -A^{-1}B\left(D - CA^{-1}B\right)^{-1} \\ -\left(D - CA^{-1}B\right)^{-1}CA^{-1} & \left(D - CA^{-1}B\right)^{-1} \end{pmatrix}.$$

Censoring and/or missing values in linear time series

We suppose that the latent process generates time series data $z_t, t = 1, \ldots, n$ and that we observe the time series $y_t, t = 1, \ldots, n$ where

$$y_t = \begin{cases} z_t & \text{if fully observed} \\ \text{NA} & \text{if censored or missing} \end{cases} \tag{3.37}$$

A secondary bivariate sequence $c_t, t = 1, \ldots, n$ indicates the status of each observation with respect to censoring,

$$c_t = \begin{cases} (c_t^-, \text{NA}) & \text{the observation is left censored with censor point } c_t^- \\ (\text{NA}, c_t^+) & \text{the observation is right censored with censor point } c_t^+ \end{cases}. \tag{3.38}$$

The t-observation is missing if $y_t = \text{NA}$ and $c_t = (\text{NA}, \text{NA})$.

The joint density function for the latent time series $(z_1, \ldots, z_n)$ may be expressed as the product the univariate conditional density functions,

$$f(z_1, \ldots, z_n) = f(z_1)\, f(z_2|z_1)\, f(z_3|z_1, z_2) \ldots f(z_n|z_1, z_2, \ldots, z_{n-1}) \tag{3.39}$$

Using the formula for the conditional mean and variance in a multivariate normal distribution it can be shown that the distribution of z_k conditional on $Z_{k-1} = (z_1 \ldots, z_{k-1})'$ is normal with mean,

$$\mu_k = \mu + (\gamma_0, \ldots, \gamma_{k-2})\, \Gamma_{k-1}^{-1}\, (Z_{k-1} - M_{k-1}) \tag{3.40}$$

where $M_k = (\mu, \ldots, \mu)'$ is the $(k-1) \times 1$ vector of unconditional means and the conditional variance is

$$\sigma_k^2 = \gamma_0 - (\gamma_0, \ldots, \gamma_{k-2})\, \Gamma_{k-1}^{-1}\, (\gamma_0, \ldots, \gamma_{k-2})'. \tag{3.41}$$

For the Expectation Step in the EM algorithm, μ_k is used if the kth observation is missing. For left-censoring with known censor point c, the expectation of a right truncated normal distribution with mean μ_k and variance σ_k is used. Using *Mathematica*, an expression for this expectation is given by

$$\mu_k^- = \frac{\mu_k\left(\text{erf}\left(\frac{c-\mu_k}{\sqrt{2}\sigma_k}\right)+1\right) - \sqrt{\frac{2}{\pi}}\sigma_k e^{-\frac{(c-\mu_k)^2}{2\sigma_k^2}}}{\text{erfc}\left(\frac{\mu_k - c}{\sqrt{2}\sigma_k}\right)} \tag{3.42}$$

where erf and erfc denote the error and complementary error functions,

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt \tag{3.43}$$

where $z \geq 0$ and $\text{erfc}(z) = 1 - \text{erf}(z)$. The error functions may also be expressed in terms of the standard normal cumulative distribution function, $\Phi(z)$,

$$\text{erf}(z) = 2\Phi\left(\sqrt{2}z\right) - 1 \tag{3.44}$$

$$\text{erfc}(z) = 2\Phi\left(-\sqrt{2}z\right) \tag{3.45}$$

Equivalent formulae for the mean of the truncated normal distribution have been derived by Barr and Sherrill [1983].

In the right-censoring case, the expectation is

$$\mu_k^+ = \frac{e^{-\frac{(c-\mu_k)^2}{2\sigma_k^2}}\left(\mu_k e^{\frac{(c-\mu_k)^2}{2\sigma_k^2}}\text{erfc}\left(\frac{c-\mu_k}{\sqrt{2}\sigma_k}\right) + \sqrt{\frac{2}{\pi}}\sigma_k\right)}{\text{erf}\left(\frac{\mu_k - c}{\sqrt{2}\sigma_k}\right)+1} \tag{3.46}$$

Algorithm CENLTSA

Initialization. Start with initial estimate of β, $\hat{\beta}^{(0)} = 0$ and for μ we may set $\hat{\mu}^{(0)}$ equal to the estimate of the mean in the case of censored normal samples. Set counter $i \longleftarrow 0$. Set maximum number of iterations, $M \longleftarrow 10^2$.

Expectation. Increment counter, $i \longleftarrow i + 1$. Compute $x_t, t = 1, \ldots, n$ where $x_t \longleftarrow y_t$ if the t-observation is fully observed. If censored $x_t \longleftarrow \mu_t^-$ or $x_t \longleftarrow \mu^+$ according as the censoring on the left or right. If missing, $x_t \longleftarrow \mu_t$. These expectations are computed using the conditional expectations where previous censored or missing values are replaced by their expectations, that $\mu_k = \mu + (\gamma_0, \ldots, \gamma_{k-2})\Gamma_{k-1}^{-1}(X_k - M_k)$, where $X_k = (x_1, \ldots, x_{k-1})'$ and M_k is the k-dimensional mean vector, $(\mu, \ldots, \mu)$. The conditional variances are given by eqn. (3.41).

Maximization. Use a suitable time series algorithm to obtain the updated estimates $\hat{\beta}^{(i)}$ and $\hat{\mu}^{(i)}$.

Convergence Test. If the estimates have converged, stop. Otherwise, return to Step 2.

In the maximization step , suitable algorithms in R include the built-in function arima() as well many others provided in some of the R packages such as arfima() [Veenstra and McLeod, 2014], FitAR() [McLeod et al., 2013]. In *Mathematica* we can use built-in function EstimatedProcess[].

EM algorithm with Durbin-Levinson algorithm

The computation of the expectation in Step 2 can be made more computationally efficient by using the Durbin-Levinson algorithm for computing the expectation for the latent process rather than recursively using eqn. (3.40) the Durbin-Levinson algorithm may be used as shown in eqn. (3.40).

Set $\phi_{1,1} = \gamma_1/\gamma_0$ and $\sigma_1^2 = \left(1 - \phi_{1,1}^2\right)\gamma_0$, where σ_k^2 denotes the variance of the k step linear predictor. Then for $k = 2, 3, ...$ we can iteratively obtain,

$$\phi_{k,k} = \left(\gamma_k - \phi_{k-1,1}\gamma_{k-1} - ... - \phi_{k-1,k-1}\gamma_1\right)/\sigma_{k-1}^2 \tag{3.47}$$

$$\begin{pmatrix} \phi_{k,1} \\ \vdots \\ \phi_{k,k-1} \end{pmatrix} = \begin{pmatrix} \phi_{k-1,1} \\ \vdots \\ \phi_{k-1,k-1} \end{pmatrix} - \phi_{k,k} \begin{pmatrix} \phi_{k-1,k-1} \\ \vdots \\ \phi_{k-1,1} \end{pmatrix} \tag{3.48}$$

and

$$\sigma_k^2 = \sigma_{k-1}^2\left(1 - \phi_{k,k}^2\right). \tag{3.49}$$

Then the conditional expectations in Step 2 are given by

$$\mu_k = \mu + \phi_{k,1}\left(x_{k-1} - \mu\right) + \cdots + \phi_{k,k}\left(x_1 - \mu\right) \tag{3.50}$$

For AR(p) models, we may use eqn. (3.40) for $k = 1, \ldots, p$ and then for $k > p$

$$\mu_k = \mu + \phi_1\left(x_{k-1} - \mu\right) + \cdots + \phi_p\left(x_1 - \mu\right) \tag{3.51}$$

Intervention analysis and regression

This algorithm can be extended to regression and intervention by replacing eqn. (3.53)

$$\mu_k = \mu - \left(\gamma_0, \ldots, \gamma_{k-2}\right)\Gamma_{k-1}^{-1}\left(X_k - M_k\right) \tag{3.52}$$

with eqn (3.53)

$$\mu_k = \mu\left(\lambda, \xi_t\right) - \left(\gamma_0, \ldots, \gamma_{k-2}\right)\Gamma_{k-1}^{-1}\left(X_k - M_k\right) \tag{3.53}$$

in Step 2, where λ is a vector of structural parameters and ξ_t are exogenous variables. For example with a dynamic pulse intervention model, $\lambda = (\mu, \omega, \delta, T)$

$$\mu\left(\lambda, \xi_t\right) = \begin{cases} \mu & t < T \\ \mu + \omega\sum_{k=0}^{t-T}\delta^k & t \geq T \end{cases} \tag{3.54}$$

Toxic water quality time series

The panel below shows running an R script using our cents package to fit CENARMA(1,1) and CENAR(1) models. Figures 3.10 and 3.11 show that Monte-Carlo tests using the Ljung-Box statistic for the two models. All scripts are included in the documentation for the cenarma() function. Each Monte-Carlo test used 1000 iterations and required less than 5 seconds. According to these portmanteau tests, the CENARMA(1,1) is adequate but not the CENAR(1).

```
> require("cents")
> Zdf <- NiagaraToxic
> z <- log(Zdf$toxic)
> iz <- c("o", "L")[1+Zdf$cQ]
> #
> #CENARMA(1,1)
> cenarma(z, iz, p=1, q=1)

Call:
arima(x = x, order = c(p, 0, q), include.mean = include.mean)

Coefficients:
         ar1      ma1  intercept
      0.9440  -0.7881    -0.9212
s.e.  0.0602   0.1157     0.1674

sigma^2 estimated as 0.2757:  log likelihood = -111.8,  aic = 231.61
> #fit CENAR(1)
> cenarma(z, iz, p=1)

Call:
arima(x = x, order = c(p, 0, q), include.mean = include.mean)

Coefficients:
         ar1  intercept
      0.2837    -0.9896
s.e.  0.0799     0.0636

sigma^2 estimated as 0.3006:  log likelihood = -117.82,  aic = 241.64
```

It should be noted that the standard errors for the estimates are should probably be a little larger. This is because these standard errors are computed under the assumption that all observations are fully observed. If there is censoring and/or missing values, the effect of this assumption would be to underestimate the true standard deviations. Because only mild censoring with a rate of about 14% was used, it seems reasonable that the standard errors of the estimates will be reasonably accurate.

We can use parametric bootstrapping to provide another estimate of the standard errors. With our general algorithm we rely on the R arima() function and this function may occasion-

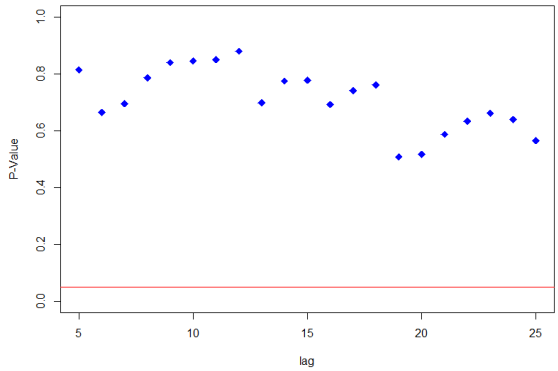

Figure 3.10: Monte-Carlo test for CENARMA(1,1).

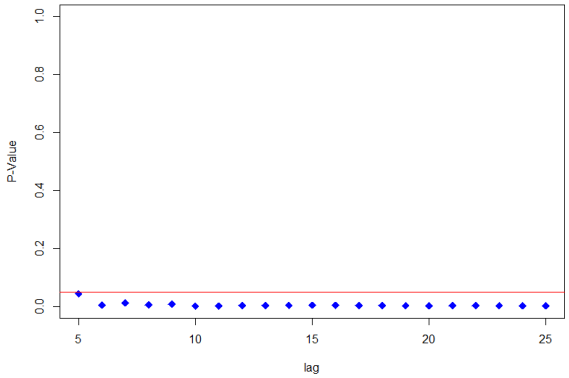

Figure 3.11: Monte-Carlo test for CENARMA(1,0).

ally produce wild results as we have seen in Figure 3.4. We did 1000 parametric bootstrap iterations and computed robust estimates of the standard deviations of the parameter estimates used the median absolute deviation specifically the following method, as suggested in Venables and Ripley [2002, §5.5] was used,

```
MAD <- function(z){
  median(abs(z-median(z)))/0.6745
}
```

The table below compares our bootstrap estimates with those produced by cenarma().

estimate	value	sd.cenarma	sd.bootstrap
$\hat{\phi}_1$	0.9440	0.0602	0.0755
$\hat{\theta}_1$	0.7881	0.1157	0.1118
$\hat{\mu}$	−0.9212	0.1674	0.1558

By comparison, assuming left-censored NID samples $\hat{\mu} = -0.9933 \pm 0.0495$. In the ARMA(0,0) case the only parameter to estimate is the mean. The required conditional expectations use the expectation result for the appropriate truncated normal distribution with parameters and so the algorithm is equivalent to the general algorithm CM presented in Chapter 2. The two algorithms provide essentially the same estimates for the mean with the toxic water quality dataset. The sample mean was $\bar{z} = -0.94$ corresponds to treating the censored values as observed values at the detection point.

One final remark comparing the time series analysis with the analysis in Chapter 2, we arrived at different models. In Chapter 2, we saw that if we assume that the level changed after the detection limit was changed then the simplest model was a level shift and about independent white noise. This model is no doubt simpler than the CENARMA(1,1) but further investigation by those involved in the collection of this data is needed to choose between these models.

Chapter 4

Conclusions

The EM algorithm is discussed for the simple case of estimation of the mean and variance in the censored time series model consists of a mean plus Gaussian white noise. This is equivalent to the well-known and much studied problem of censored samples from a normal distribution [Cohen, 1991, Schneider, 1986, Wolynetz, 1979a]. We present a new derivation using the EM algorithm as well a new closed form expression for the information matrix for the mean and variance parameters. It is shown that in the censored case these parameters are not orthogonal. A new interactive normal probability plot for censored data is discussed. Several applications are given.

In addition, we develops a new Quasi-EM algorithm for fitting ARMA, stationary ARFIMA and other linear time series models to censored time series. It is shown that the missing value problem in time series model fitting may be regarded as a special and extreme case of censoring and it is demonstrated that our approximate Quasi-EM algorithm handles this case just as well as the standard exact treatment. The method is illustrated with an application.

In future research, we plan to develop specialized two-sample tests and confidence intervals for censored random samples and to extend the methods to the family of exponential distributions, multivariate and spatial time series. I am currently preparing paper with my supervisor Dr. McLeod to deal with the problems of how large a sample size is needed to determine a confidence interval of a certain width given preliminary estimates of the parameters? How large a sample size is needed to achieve some specified power of a statistical test?

Bibliography

C. F. Ansley. An algorithm for the exact likelihood of a mixed autoregressive-moving average process. *Biometrika*, 66(1):59–65, 1979. doi: 10.1093/biomet/66.1.59.

Donald R. Barr and E. Todd Sherrill. On the convergence of the EM algorithm. *The American Statistician*, 53(4):pp. 357–361, 1983.

G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Wiley, 2008.

Russell A. Boyles. On the convergence of the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 45(1):pp. 47–50, 1983. URL `http://www.jstor.org/stable/2345622`.

G. Casella and R.L. Berger. *Statistical inference*. Thomson Learning, 2002.

A. C. Cohen. *Truncated and Censored Samples: Theory and Applications*. Dekker, 1991.

A. C. Davison and D. V. Hinkley. *Bootstrap Methods and Their Applications*. Cambridge University Press, Cambridge, 1997.

J. Durbin and S.J. Koopman. *Time Series Analysis by State Space Methods*. Oxford, 2nd edition, 2012.

Bradley Efron and David V. Hinkley. Assessing the accuracy of the maximum likelihood estimator: Observed versus expected fisher information. *Biometrika*, 65(3):pp. 457–482, 1978. URL `http://www.jstor.org/stable/2335893`.

Abdelhameed El-Shaarawi, Nagham Muslim Mohammad, and Ian McLeod. Comparing gamma and log-normal distributions, 8 2013. URL `http://demonstrations.wolfram.com/`.

Stephen J. Finch, Nancy R. Mendell, and Jr. Thode, Henry C. Probabilistic measures of adequacy of a numerical search for a global maximum. *Journal of the American Statistical Association*, 84(408):pp. 1020–1023, 1989. URL `http://www.jstor.org/stable/2290078`.

A. K. Gupta. Estimation of the mean and standard deviation of a normal population from a censored sample. *Biometrika*, 39:pp. 260–273, 1978.

D. R. Helsel. *Statistics for Censored Environmental Data Using Minitab and R*. Wiley, 2nd edition, 2011.

A. Henningsen. *censReg: Censored Regression (Tobit) Models*, 2012. URL http://CRAN.R-project.org/package=censReg. R package version 0.5-20. Date: 2013/08/20. Access Date May 20, 2014.

Arne Henningsen and Ott Toomet. maxlik: A package for maximum likelihood estimation in R. *Computational Statistics*, 26(3):443–458, 2011.

Philip K. Hopke, Chuanhai Liu, and Donald B. Rubin. Multiple imputation for multivariate data with missing and below-threshold measurements: Time-series concentrations of pollutants in the arctic. *Biometrics*, 57(1):pp. 22–33, 2001. URL http://www.jstor.org/stable/2676838.

Richard H. Jones. Maximum Likelihood Fitting of ARMA Models to Time Series with Missing Observations. *Technometrics*, 22(3):389–395, 1980. URL http://dx.doi.org/10.2307/1268324.

Kaplan and Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:45781, 1958.

Yukio Kasahara, Mohsen Pourahmadi, and Akihiko Inoue. Duals of random vectors and processes with applications to prediction problems with missing values. *Statistics & Probability Letters*, 79(14):1637–1646, 2009.

K. Knight. *Mathematical Statistics*. CRC Press, 2000.

Robert Kohn and Craig F. Ansley. Estimation, prediction, and interpolation for arima models with missing data. *Journal of the American Statistical Association*, 81(395):pp. 751–761, 1986.

J. F. Lawless. *Statistical Models and Methods for Lifetime Data*. Wiley, 2003.

Greta M. Ljung. The likelihood function for a stationary gaussian autoregressive-moving average process with missing observations. *Biometrika*, 69(1):pp. 265–268, 1982.

Greta M. Ljung and G. E. P. Box. The likelihood function of stationary autoregressive-moving average models. *Biometrika*, 66:pp. 265–270, 1979.

G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 2nd edition, 2007.

A. I. McLeod and Nagham Muslim Mohammad. *Censored time series analysis*, 2012. URL http://CRAN.R-project.org/package=censReg. R package version 0.5-20. Date: 2013/08/20. Access Date May 20, 2014.

A.I. McLeod, Hao Yu, and Z. Krougly. Algorithms for linear time series analysis: With r package. *Journal of Statistical Software*, 23(5):1–26, 2007. URL http://www.jstatsoft.org/v23/i05.

A.I. McLeod, Hao Yu, and Z. Krougly. *ltsa: Linear time series analysis*, 2012. URL http://CRAN.R-project.org/package=ltsa. R package version 1.4.4. Access date: 2014/07/30.

A.I. McLeod, Ying Zhang, and Changjiang Xu. *FitAR: Subset AR Model Fitting*, 2013. URL http://CRAN.R-project.org/package=FitAR. R package version 1.94. Access date: 2014/07/30.

Ian McLeod and Nagham Muslim Mohammad. Monte carlo expectation-maximization algorithm, 7 2013a. URL http://demonstrations.wolfram.com/.

Ian McLeod and Nagham Muslim Mohammad. Comparing exact and approximate censored normal likelihoods, 7 2013b. URL http://demonstrations.wolfram.com/.

Nagham Muslim Mohammad and Ian McLeod. Estimating and diagnostic checking in censored normal random samples, 8 2013. URL http://demonstrations.wolfram.com/.

F.H. Nieto and J. Martinex. Interpolation, outliers and inverse autocorrelations. *Communications in Statistics (Theory and Methods)*, 20:3175–3186, 1991.

F.H. Nieto and J. Martinex. A recursive approach for estimating missing observations in univariate time series. *Communications in Statistics (Theory and Methods)*, 25(9):2101–2116, 1996.

David Oakes. Direct calculation of the information matrix via the em algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(2):pp. 479–482, 1999. URL http://www.jstor.org/stable/2680653.

Jung Wook Park, Marc G. Genton, and Sujit K. Ghosh. Estimation and forecasting of long-memory processes with missing values. *Journal of Forecasting*, 16:pp. 395–410, 1997.

Jung Wook Park, Marc G. Genton, and Sujit K. Ghosh. Censored time series analysis with autoregressive moving average models. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 35(1):pp. 151–168, 2007. URL http://www.jstor.org/stable/20445244.

Jeremy Penzer and Brian Shea. The exact likelihood of an autoregressive-moving average model with incomplete data. *Biometrika*, 84(4):pp. 919–928, 1997.

C. R. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2004.

Christian Robert. Simulation of truncated normal variables. *Statistics and Computing*, 5(2): 121–125, 1995.

H. Schneider. *Truncated and Censored Samples from Normal Populations*. Dekker, 2nd edition, 1986.

R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications*. Springer, 2000.

Justin Q. Veenstra and A. I. McLeod. *arfima: Fractional ARIMA Time Series Modeling*, 2014. URL http://CRAN.R-project.org/package=arfima. R package version 1.2-6. Access date: 2014/07/31.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, 4th edition, 2002.

M. S. Wolynetz. Algorithm as 138: Maximum likelihood estimation from confined and censored normal data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28 (2):pp. 185–195, 1979a. URL http://www.jstor.org/stable/2346748.

M. S. Wolynetz. Algorithm as 139: Maximum likelihood estimation in a linear model from confined and censored normal data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(2):pp. 195–206, 1979b. URL http://www.jstor.org/stable/2346749.

M. S. Wolynetz. Remark as r31: A remark on algorithm as 139. maximum likelihood estimation in a linear model from confined and censored normal data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(2):p. 228, 1980. URL http://www.jstor.org/stable/2986317.

M. S. Wolynetz. Remark as r32: A remark on algorithm as 139: Maximum likelihood estimation in a linear model from confined and censored normal data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 30(1):p. 105, 1981. URL http://www.jstor.org/stable/2346671.

C. F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11 (1):pp. 95–103, 1983. URL http://projecteuclid.org/download/pdf_1/euclid.aos/1176346060.

Ying Ying Zhang, Hao Yu Hao, and A. I. McLeod. Developments in maximum likelihood unit root tests. *Communications in Statistics - Simulation and Computation*, 42(5):pp. 1088–1103, 2013.

Curriculum Vitae

Publications:

- A. I. McLeod and N. M. Mohammad, (2014). cents: Censored Time Series. R package, Version 0.1-41. http://cran.r-project.org/web/packages/cents/cents.pdf.
- A. I. McLeod and N. M. Mohammad, (2014). CensNID: Censored NID Samples. R package, Version 0-0-1. http://cran.r-project.org/web/packages/censNID/censNID.pdf
- Ian McLeod and Nagham Muslim Mohammad. Monte carlo expectation-maximization algorithm, 7 2013a. URL http://demonstrations.wolfram.com/.
- Ian McLeod and Nagham Muslim Mohammad. Comparing exact and approximate censored normal likelihoods, 7 2013b. URL http://demonstrations.wolfram.com/.
- Nagham Muslim Mohammad and Ian McLeod. Estimating and diagnostic checking in censored normal random samples, 8 2013. URL http://demonstrations.wolfram.com/.
- Ian McLeod and Nagham Muslim Mohammad. Comparing exact and approximate censored normal likelihoods,7 2013. URL http://demonstrations.wolfram.com/.
- Ian McLeod and Nagham Muslim Mohammad. Transformation to Symmetry of Gamma Random Variables, 6 2013. URL http://demonstrations.wolfram.com/.

Conference Presentations:

- 42nd Annual Meeting of the Statistical Society of Canada, University of Toronto, May 25 - 28, 2014. Ian McLeod and Nagham Muslim Mohammad. Censored Time Series Analysis.
- Interdisciplinary AMMCS Conference Series. Waterloo, Ontario, Canada, 2013. Nagham Muslim Mohammad, Abdelhameed El-Shaarawi and Ian McLeod. Censored Gamma Regression with Applications.
- 40th Annual Meeting of the Statistical Society of Canada, University of Guelph, June 3-6, 2012. Ian McLeod and Nagham Muslim Mohammad. Left-Censored Gamma.
- 9th Annual Earth Day Colloquium, University of Western Ontario, April 12 - 13, 2012. Ian McLeod and Nagham Muslim Mohammad. Censored Gamma Regression for Forest Fire prediction

Name: Nagham Muslim Mohammad

Post-Secondary Education and Degrees:

The University of Western Ontario
London, Ontario, Canada
2009 - 2014 Ph.D

University of Western Ontario
London, ON, Canada
2007 - 2009 M.Sc

Baghdad University
Baghdad, Iraq
1991-1993 M.Sc.

Baghdad University
Baghdad, Iraq
1991-1993 B.Sc.

Honours and Awards:

Western Graduate Research Scholarship
The University of Western Ontario
2007-2014

University Employment History:

Assistant Professor/Lecturer
McMaster University
Hamilton, Ontario, Canada, 2013-2014

Lecturer
The University of Western Ontario
London, Ontario, Canada, 2012-2013

Teaching Assistant
The University of Western Ontario
London, Ontario, Canada, 2007-2012

Research Assistant
The University of Western Ontario
London, Ontario, Canada, 2007-2014

Assistant Professor
Al-Fatah University
Tripoli, Libya, 1997-2005

Assistant Professor
Baghdad University
Baghdad, Iraq, 1994-1997

Assistant Programmer
Baghdad University
Baghdad, Iraq, 1988-1991